

Phylogenetic Networks that Display a Tree Twice

A thesis
submitted in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy
in Mathematics

By Paul Joseph Cordue



The University of Canterbury
School of Mathematics and Statistics

2016

Contents

Acknowledgments	1
Abstract	2
1 Introduction	3
1.1 Overview	6
1.2 Basic Definitions and Lemmas that are Used Throughout the Thesis	6
1.3 Switchings and Avoidability	10
2 Displaying Exactly p Trees with $p - 1$ Reticulation Vertices	14
2.1 Introduction	14
2.2 The Construction	14
3 Answering the Driving Question of this Thesis for a Particular Class of Networks	17
3.1 Introduction	17
3.2 Avoidable Cycles	19
3.3 Characterisation	21
3.4 Quadratic-Time Algorithm	25
3.5 Remark on Tree-Child and Normal Networks	29
4 Characterising when a Phylogenetic Network Displays a Tree Twice in Terms of Substructures in the Network	31
4.1 Introduction	31
4.2 Basic Networks and Basic Cyclic Pairs	32
4.3 Representing a Basic Cyclic Pair	34
4.3.1 What does it mean for the representation of a basic cyclic pair to display a tree twice?	35
4.4 Reducing a Basic Cyclic Pair	39

4.5	Avoidable Reticulation Vertices	42
4.6	Characterising when a Phylogenetic Network Displays a Tree Twice in Terms of Cyclic Pairs in the Network	43
4.6.1	Associated basic cyclic pairs	47
4.6.2	Processed cyclic pairs	50
5	The Representation of a Basic Cyclic Pair	53
5.1	Introduction	53
5.2	Characterising when the Representation of a Basic Cyclic Pair Displays a Tree Twice	55
5.3	Characterising when a Basic ⁰ Representation Displays a Tree Twice	60
5.3.1	Case 1: Both $R_{\mathcal{T}_2}$ and $W_{\mathcal{T}_1}$ are empty	60
5.3.2	Case 2: Exactly one of $R_{\mathcal{T}_2}$ and $W_{\mathcal{T}_1}$ is empty	61
5.3.3	The final case	62
5.3.3.1	Dropping some assumptions	71
5.3.3.2	An iterative algorithm	73
5.3.3.3	How the iterative algorithm works	76
5.3.3.4	Running-time of the iterative algorithm	89
5.4	A Sufficient Condition	91
6	An Algorithm for Deciding whether or not a Basic Cyclic Pair Displays a Tree Twice	100
6.1	Introduction	100
6.2	An Efficient Algorithm for Deciding whether or not a Basic ⁰ Representation Displays a Tree Twice	100
6.2.1	Proof that BASIC ⁰ CHECKER works	101
6.2.2	Running-time of BASIC ⁰ CHECKER	103
6.3	Checking a Basic Cyclic Pair	103
6.3.1	Desirable configuration algorithm	104
6.3.1.1	Proof that DESIRABLE CONFIG works	105
6.3.2	A representation that needs further checking	108
6.3.3	Checking a representation that needs further checking	109
6.3.4	Basic cyclic pair algorithm	113

7	An Efficient Algorithm for Deciding whether or not a Phylogenetic Network Displays a Tree Twice	115
7.1	Introduction	115
7.2	Checking whether or not a Network has any Avoidable Reticulation Vertices	116
7.2.1	A bound on the number of vertices in a network without any avoidable reticulation vertices nor any trivial cyclic pairs	116
7.2.2	An algorithm for deciding whether or not a network has an avoidable reticulation vertex	121
7.3	Finding Cyclic Pairs that Need Checking	123
7.4	The Efficient Algorithm	125
8	A Class of Locally Independent Networks	128
8.1	Introduction	128
8.2	Definition and Properties of a Locally Independent Network	128
8.3	An Equivalence Relation	130
8.4	Algorithm	131
	Bibliography	134

Acknowledgments

First and foremost, I would like to acknowledge the support and guidance from Charles Semple, Simone Linz, and Mike Steel; I would not have got this far without them. I would also like to acknowledge the financial support from the Marsden Fund via Charles and Mike, as well as the financial support from the 7th European Community Framework Programme that got me over to the University of Tübingen, Germany, for a few weeks via Simone. Thanks to Geoff Whittle for putting me in contact with Charles and for having me as a visitor, honorary matroid theorist, and friend whenever I was in Wellington.

There are many others that helped me throughout my time as a Ph.D. student at the University of Canterbury, both administratively and socially, so thanks to the University of Canterbury and the School of Mathematics and Statistics. Of course, a big thank you to my parents, family, and friends for everything else.

Abstract

In the study of phylogenetics, which is the study of how forms of life evolve and relate to each other, there is great scope for mathematics to get involved. One such study of phylogenetics that currently employs mathematics is the study of phylogenetic networks and phylogenetic trees. Phylogenetic networks and trees can be used to represent how life evolved with the former having the ability to represent biological processes such as hybridization, horizontal gene transfer, and gene recombination. In terms of mathematics, one sees phylogenetic networks and trees as directed graphs. A phylogenetic network N displays a rooted phylogenetic tree T if all of the ancestral history inferred by T is also inferred by N . The main result of this thesis is a quartic-time, in terms of the number of leaves in the network, algorithm that decides whether or not a given phylogenetic network displays a tree twice. As a consequence of the work leading to the main result, a class of phylogenetic networks is discovered such that there is a quadratic-time, in terms of the number of leaves in the network, algorithm for counting the number of distinct trees displayed by a given network in the class. These results are interesting because it has been shown that in general counting the number of trees displayed by a given phylogenetic network is $\#P$ -complete. Thus the main result of this thesis opens the door to insights regarding a computationally hard problem.

Chapter 1

Introduction

Phylogenetics is the reconstruction and analysis of phylogenetic (evolutionary) trees and networks based on inherited characteristics [1]. The tools and techniques of phylogenetics have been used in fields such as evolutionary biology, linguistics, cancer research, epidemiology, virology, and conservation biology. With the help of phylogenetics a number of multifaceted problems have been investigated and resolved, such as the origin of the HIV virus [1]. Phylogenetics has even been used to reconstruct the copying history of manuscripts (see [2]). One of the many tools of phylogenetics is the phylogenetic tree or evolutionary tree. Since the time of Charles Darwin's *The Origin of Species* [3], phylogenetic trees have played an important role in the study of evolution, because of a phylogenetic tree's ability to represent speciation events. It is now becoming apparent that representing evolution with a tree is not sufficient, because of the presence of reticulation events. Reticulation events include hybridization, horizontal gene transfer, and DNA recombination. DNA recombination involves the exchange of genetic material either between multiple chromosomes or between different regions of the same chromosome [4]. Horizontal (lateral) gene transfer is the transfer of genes between different species [5]. Whilst hybridization can be defined as reproduction between members of genetically distinct populations producing offspring of mixed ancestry [6]. About 10-30% of multicellular animal and plant species hybridize regularly [6]. Hybridization also played an important role in the evolution of bread wheat, because the findings from [7] imply that the present-day bread wheat genome is a product of multiple rounds of hybrid speciation [7]. Hence evolution is more accurately represented by an entwined network that can represent both speciation events and reticulation events (for some books on phylogenetics see [1], [8], [9], [10], [11], [12]).

Although phylogenetic networks are becoming increasingly important in studying the evolution of present-day species whose past includes reticulation events, phylogenetic trees continue to play a fundamental role in phylogenetic analyses since the evolutionary history of a single gene can, in

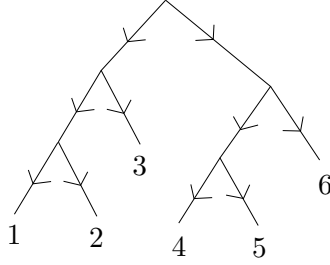


Figure 1.0.1: The figure shown is an example of a phylogenetic tree. The leaves are labeled 1, 2, 3, 4, 5, 6.

most cases, be described by a tree. It is therefore not surprising that investigating the tree-like content of phylogenetic networks is often an important first step in analyzing and interpreting such networks. For example, one might be interested in deciding if a phylogenetic network displays a given phylogenetic tree or in counting the number of trees displayed by a network. The latter problem is related to calculating the parsimony score of a network [13] which, given the popularity of parsimony tree reconstruction algorithms, is likely to become a standard tool in computing a phylogenetic network directly from sequence data. While deciding if a tree is displayed by a network is polynomial-time solvable for certain special classes of phylogenetic networks (for the work done on the aforementioned problem see [14], [15], [16], [17], [18]), the problem is NP-complete in its general form [19]. Similarly, counting the number of phylogenetic trees that are displayed in an arbitrary phylogenetic network is also known to be a computationally hard problem [20].

A simpler problem relative to counting the number of phylogenetic trees displayed by a network is deciding if a network displays the maximum number of trees, which, of course, is the same as deciding whether or not a network displays a tree twice. An example of when one needs to have a phylogenetic tree displayed more than once in a phylogenetic network is in the case where a phylogenetic tree is sufficient for the representation of a species' evolution but there is some uncertainty about how the present-day descendants are related to one another or how they evolved from their ancestors; hence a phylogenetic network can be used in the absence of reticulation events, when there is some uncertainty in the true (tree-shaped) phylogeny [21]. If there are a number of plausible ways in which the present-day descendants could be related to each other (represented by a number of distinct phylogenetic trees) then all the information contained in those phylogenetic trees can be held in a single phylogenetic network. It is known that if a phylogenetic network N has k reticulation vertices (vertices that have two edges going in and one edge going out), then the maximum number of possible trees displayed by N is 2^k , because for each reticulation vertex v in N there are two possible ways to enter v . So if we want a phylogenetic network to display a number

of phylogenetic trees where that number is not 2^k , then the phylogenetic network has to display a tree twice. This means that there exists two different ways of choosing exactly one reticulation edge (an edge going into a reticulation vertex) for each reticulation vertex in the network, and the result of these two different ways of choosing reticulation edges is one tree. Therefore one may want to ask “When does a phylogenetic network display a tree twice?” This question drives this thesis. The challenge is to be able to decide in polynomial-time whether or not a given phylogenetic network displays a tree twice.

Another question that is answered in this thesis is: “Given a positive integer n , does there exist a phylogenetic network that displays exactly n distinct trees?” That question is motivated by the work done in [20], where the authors showed (as a by-product of their main result) that, given a Fibonacci number m , there exists a phylogenetic network that displays exactly m distinct phylogenetic trees. By constructing a phylogenetic network that displays exactly n distinct trees answers the question “Given a positive integer n , does there exist a phylogenetic network that displays exactly n distinct trees?”

Answering the driving question of this thesis is a much more involved process, and is outlined as follows: We characterise when a phylogenetic network displays a tree twice in terms of a local substructure inherent in the network. We then show that this local substructure can be represented by a phylogenetic tree, and that one can go from the local substructure to its tree representation and back again without losing any information, relative to our aim. We work on the tree representation and translate what it means for a representation to display a tree twice. The translation from networks to trees means that we start working with set operations such as intersection, union, and set minus, as well as the tree operation of leaf restriction. That translation makes it easier to navigate the complexities that arise when characterising when the local substructure displays a tree twice. From there we develop a polynomial-time algorithm for deciding whether or not the local substructure displays a tree twice based on the characterisation. Lastly, this leads to a polynomial-time algorithm for deciding whether or not the network displays a tree twice. Hence the main result of this thesis is a polynomial-time algorithm for deciding whether or not a given phylogenetic network displays a tree twice. Consequent to the work leading to the main result, it is also revealed that there is a class of networks for which the number of distinct trees displayed by a network in the class can be computed in polynomial-time with the help of the multiplication principle.

1.1 Overview

Chapter 2 shows how to construct a phylogenetic network that displays exactly n trees, for a given positive integer n . Chapter 3 characterises when a particular class of phylogenetic networks display a tree twice and shows that there is a polynomial-time algorithm for deciding whether or not any network in the particular class displays a tree twice. The characterisation of when a phylogenetic network displays a tree twice in terms of a local substructure in the network is in Chapter 4, as well as how to represent the local substructure as a phylogenetic tree. In Chapter 5 we characterise when the representation displays a tree twice. In Chapter 6 we present a polynomial-time algorithm for deciding whether or not the representation of the local substructure displays a tree twice based on the work in Chapter 5. Now Chapter 7 is when all the work from Chapters 4, 5, and 6 come together to give a polynomial-time algorithm for deciding whether or not a phylogenetic network displays a tree twice. Chapter 8 describes a class of networks such that counting the number of distinct trees displayed by any network in the class can be done in polynomial-time.

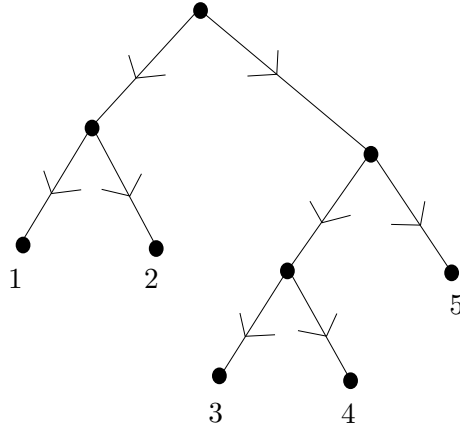
Except where duly and clearly noted, the results of Chapters 2, 4, 5, 6, 7, and 8 are new. The third chapter is based on the paper [22] that was done jointly with Charles Semple and Simone Linz.

1.2 Basic Definitions and Lemmas that are Used Throughout the Thesis

This section provides notation and terminology that is used throughout the thesis. Throughout the thesis, X denotes a finite set.

Phylogenetic trees. A *rooted phylogenetic X -tree* T is a rooted tree in which the root has degree at least two and all other interior vertices have degree at least three, and whose leaf set is X . In addition, T is *binary* if, apart from the root which has degree two, all interior vertices have degree three. Since we are interested only in rooted binary phylogenetic X -trees throughout the thesis, we will almost always refer to such a tree as a *tree on X* .

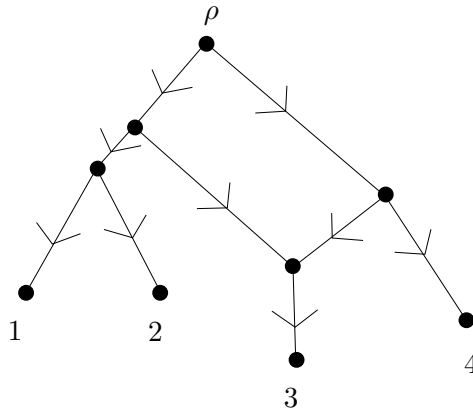
Example 1.1. The following figure is an example of a rooted binary phylogenetic X -tree, where $X = \{1, 2, 3, 4, 5\}$.



Phylogenetic networks. A *phylogenetic network* N on X is a rooted acyclic digraph that satisfies the following three properties:

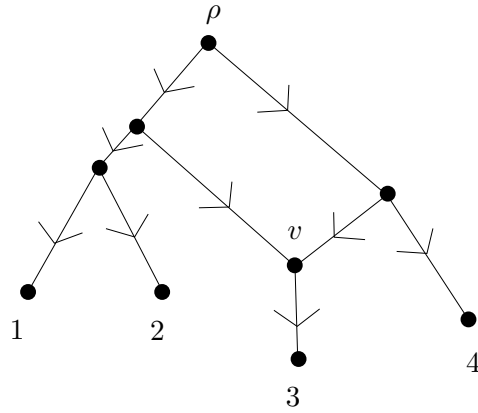
- (i) the root has out-degree two,
- (ii) each vertex with out-degree zero has in-degree one, and the set of vertices with out-degree zero is X , and
- (iii) all other vertices either have in-degree one and out-degree two, or in-degree two and out-degree one.

Example 1.2. The following figure is a phylogenetic network on X , where $X = \{1, 2, 3, 4\}$. The root is labeled ρ .

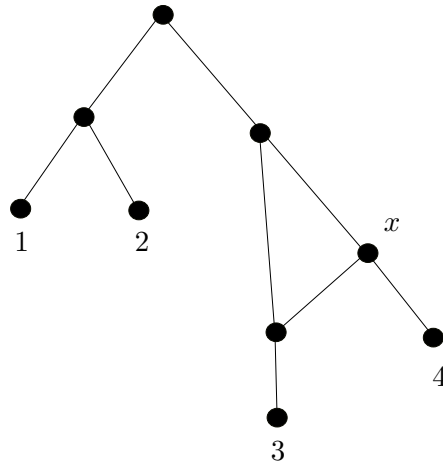


We will refer to N as a *network on X* or, simply, as a *network* if X plays no particular role. Such networks are commonly referred to as *binary* phylogenetic networks. Here, as well as in all other figures, edges are directed down the page. Furthermore, we will assume that networks have no parallel edges. For a network N , vertices with in-degree two and out-degree one are called *reticulation vertices* or *reticulations* and all other vertices are called *tree vertices*.

Example 1.3. A phylogenetic network with its only reticulation vertex labeled v .

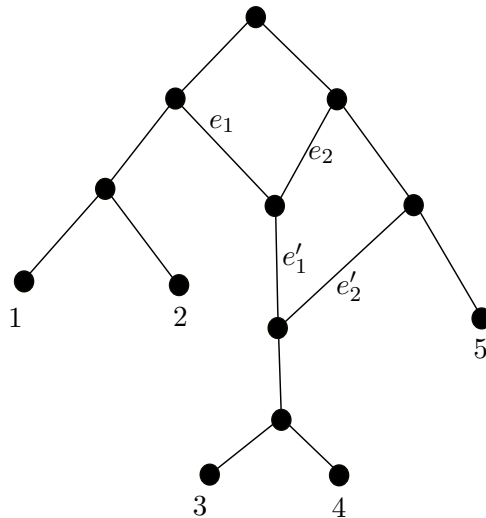


Example 1.4. A phylogenetic network with one of its tree vertices labeled x .



In addition, edges directed into a reticulation are called *reticulation edges* and all other edges are called *tree edges*. Similar to rooted phylogenetic trees, vertices with out-degree zero are referred to as *leaves*. Indeed, a rooted binary phylogenetic tree is a phylogenetic network with no reticulations.

Example 1.5. The edges/arcs that are marked are reticulation edges/arcs.

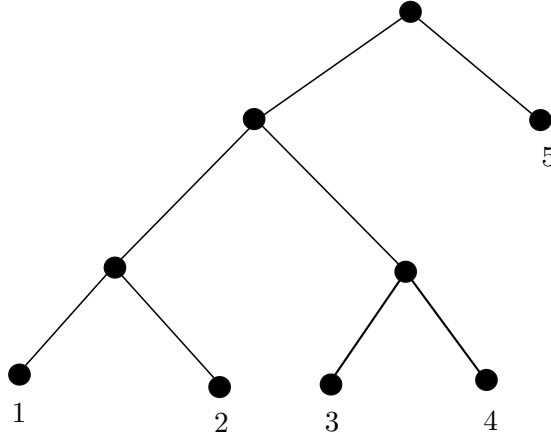


Biologically, like phylogenetic trees, phylogenetic networks illustrate the evolutionary history of a collection of present-day species. Such species are represented by the leaves, while all other vertices represent (hypothetical) ancestors. A reticulation represents, for example, a hybrid species.

Let u and v be two vertices of a network N on X . If there is a directed path (resp. a directed path that contains at least one edge) from u to v , then u is an *ancestor* (resp. *strict ancestor*) of v , and v is a *descendant* (resp. *strict descendant*) of u . More particularly, if (u, v) is an edge in N , then u is a *parent* of v , and v is a *child* of u . Furthermore, if two vertices have a common parent, then they are said to be *siblings*.

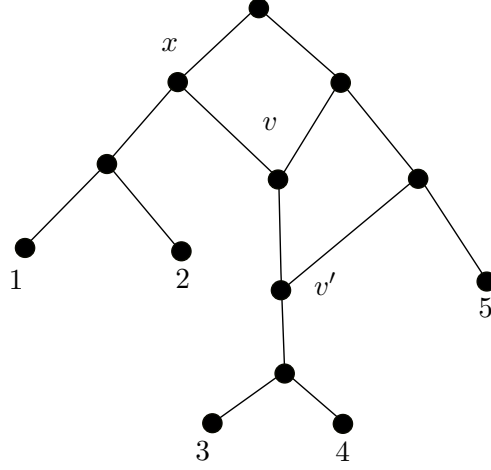
Let T be a tree on X , and let N be a network on X . We say that N *displays* T if T can be obtained from N by deleting edges and vertices, and contracting vertices with in-degree one and out-degree one. Intuitively, T is displayed by N if all of the ancestral information inferred by T is also inferred by N . Note that if T is displayed by N , then T is necessarily binary.

Example 1.6. The tree shown can be obtained from the network in Example 1.5 by deleting the reticulation edges e_2 and e'_2 then suppressing all vertices with in-degree one and out-degree one.



Definition 1.1. Let N be a phylogenetic network. A directed path P is called a *tree-path* if every vertex, except for the first and last, on P is a tree-vertex.

Example 1.7. The directed path from v to v' is a tree-path. The only directed path from x to v' through v is not a tree-path. The only directed path from v to leaf 4 is not a tree-path. The only directed path from v' to leaf 4 is a tree-path.



1.3 Switchings and Avoidability

In this section, we introduce the concept of switchings in a network to describe precisely what it means for a tree to be displayed twice. We also introduce the concept of avoidability, which is crucial in characterising when a network displays a tree twice.

Switchings. Let N be a network on X . A subset S of reticulation edges of N is a *switching* of N if, for each reticulation v of N , the set S contains precisely one of the two reticulation edges directed into v .

Example 1.8. The set $S = \{e_1, e'_1\}$ is a switching of the network shown in Example 1.5, whilst the set $G = \{e_1, e'_1, e'_2\}$ is not a switching of the network shown in Example 1.5. Likewise the set $B = \{e_1\}$ is not a switching of the network shown in Example 1.5.

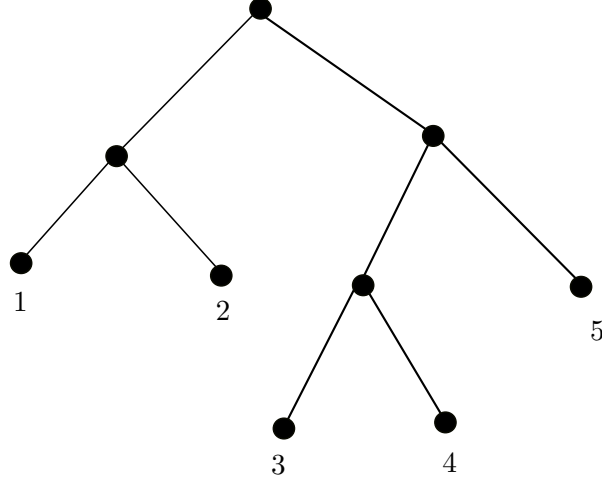
Now, let S be a switching of N . If we delete each reticulation edge in N that is not in S , then the resulting directed graph contains no underlying cycle and, for each leaf $\ell \in X$, it is easily checked that there is a directed path from the root of this directed graph to ℓ . If we now repeatedly contract each resulting vertex with in-degree one and out-degree one and delete each degree one vertex that is not in X , it is easily seen that we obtain a tree T on X . We say that S *yields* T . Note that T is well-defined and, by construction, T is displayed by N . Conversely, observe that, if T is a tree on X displayed by N , then there exists a switching that yields T . In summary, this leads to the following lemma, which we will freely use throughout the thesis.

Lemma 1.1. *A network N on X displays a tree T on X if and only if there exists a switching S of N that yields T .*

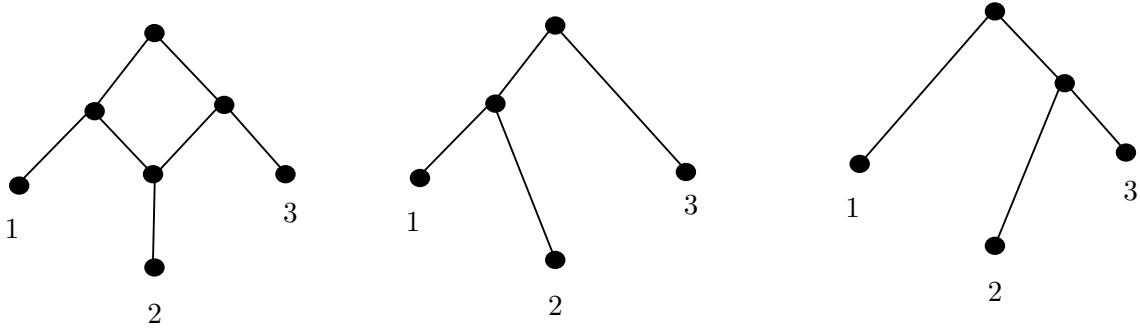
Example 1.9. The switching $S = \{e_1, e'_1\}$ applied to the network shown in Example 1.5 yields the tree shown in Example 1.6.

With Lemma 1.1 in hand, we say that N *displays a tree twice* if there exists two distinct switchings of N each of which yields (up to phylogenetic isomorphism, i.e. up to relabeling internal vertices and swapping leaves that share the same parent) the same tree on X . Referring back to a comment made in the introduction, it follows from Lemma 1.1 that if N is a network on X with exactly k reticulations, then N displays at most 2^k distinct trees on X .

Example 1.10. The two distinct switchings $S_1 = \{e_1, e'_2\}$ and $S_2 = \{e_2, e'_2\}$ applied to the network shown in Example 1.5 yield the following tree.

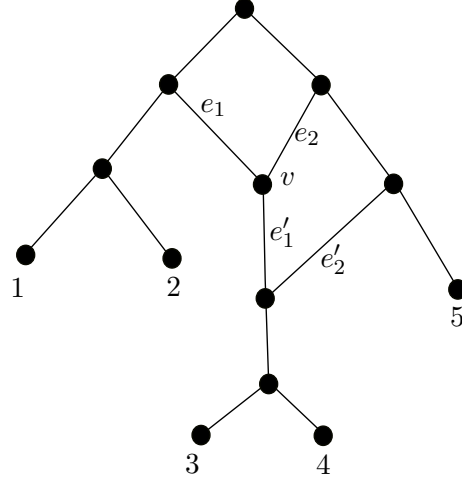


Example 1.11. The network on the left yields the two trees to the right of the network. Those trees are isomorphic in the graph theory sense but they are not isomorphic in the phylogenetic sense.



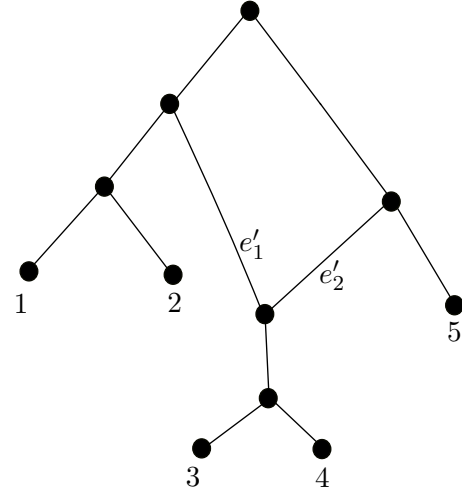
Definition 1.2. A subset G of reticulation edges of N is a *general switching* of N when for each reticulation vertex v of N either exactly one of v 's reticulation edges is in G or both of v 's reticulation edges are in G .

Example 1.12. The set $G = \{e_1, e'_1, e'_2\}$ is a general switching of the network shown below. The set $B = \{e_1\}$ is not a general switching of the network shown below. The set $S_2 = \{e_2, e'_2\}$ is both a general switching and a switching of the network shown below.



Definition 1.3. If G is a general switching of a network N and we delete each reticulation edge in N that is not in G , then for each leaf $\ell \in X$, it is easily checked that there is a directed path from the root of this directed graph to ℓ . If we now repeatedly suppress each resulting vertex with in-degree one and out-degree one and delete each degree one vertex that is not in X , then we obtain a network N' on X . We say that G *yields* N' .

Example 1.13. The general switching $G = \{e_1, e'_1, e'_2\}$ applied to the network shown in Example 1.12 yields the following network.

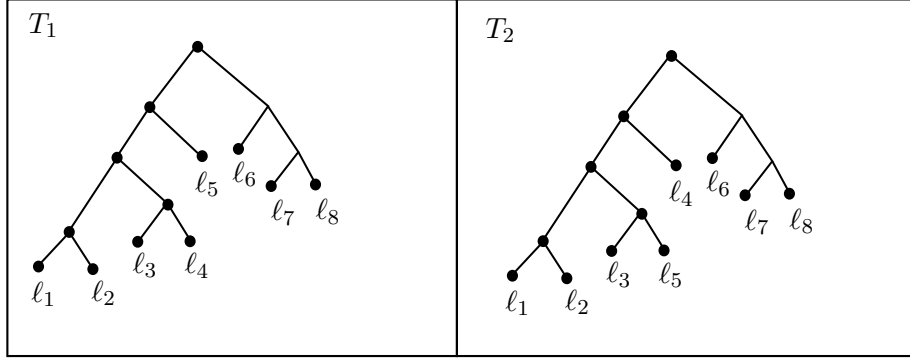


Definition 1.4. Let N and N' be networks on X . We say that N *displays* N' when there exists a general switching G of N such that G yields N' .

The next definition leads to a lemma that characterises when two trees are distinct.

Definition 1.5. Let T be a tree on leaf set X , and let $\ell_1, \ell_2, \ell_3 \in X$. We say that T *contains the triple* $\ell_1\ell_2|\ell_3$ (or equivalently $\ell_2\ell_1|\ell_3$) when the path connecting ℓ_1 and ℓ_2 does not intersect the path from the root of T to ℓ_3 .

Example 1.14. The tree T_1 on the left contains the triple $\ell_1\ell_3|\ell_5$. The tree T_2 on the right does not contain the triple $\ell_1\ell_3|\ell_5$, but it does contain the triple $\ell_3\ell_5|\ell_1$. Hence we see that $T_1 \neq T_2$.



Lemma 1.2. Let T_1 and T_2 be two trees on the leaf set X . The two trees T_1, T_2 are distinct if and only if there exists $\ell_1, \ell_2, \ell_3 \in X$ such that either T_1 contains the triple $\ell_1\ell_2|\ell_3$ whilst T_2 does not, T_1 contains the triple $\ell_1\ell_3|\ell_2$ whilst T_2 does not, or T_1 contains the triple $\ell_2\ell_3|\ell_1$ whilst T_2 does not.

Avoidable vertices. Let N be a network on X , and let v be a vertex of N . We say that v is *avoidable* if, for each $\ell \in X$, there exists a directed path from the root of N to ℓ that avoids v . Otherwise, v is *unavoidable*. In particular, if v is unavoidable, then there exists a leaf ℓ such that every directed path from the root of N to ℓ contains v . To illustrate, Example 1.12 shows a network with an avoidable reticulation v . Note that the definition of an unavoidable reticulation coincides with that of a so-called *visible* reticulation in [8].

The next lemma gives a sufficient, but not a necessary, condition for guaranteeing that a network displays a tree twice.

Lemma 1.3. Let N be a network on X . If N has an avoidable reticulation, then N displays a tree on X twice.

Proof. Let v be an avoidable reticulation of N , and let e_1 and e_2 be the two reticulation edges that are incident with v . Since v is avoidable, there exists, for each $\ell \in X$, a directed path P_ℓ from the root of N to ℓ that avoids v . Let T be a tree on X that is displayed by N and, up to degree two vertices, whose edge set is a subset of $\bigcup_{\ell \in X} P_\ell$. It is easily seen that such a T always exists. Now, let S be a switching of N that yields T . It follows that the two distinct switchings $(S - \{e_1, e_2\}) \cup \{e_1\}$ and $(S - \{e_1, e_2\}) \cup \{e_2\}$ both yield T and, hence, N displays a tree on X twice. \square

Chapter 2

Displaying Exactly p Trees with $p - 1$ Reticulation Vertices

2.1 Introduction

This chapter is motivated by the work done in [20]. The authors of [20] positively answered, as a by-product of their main result, the following question: “Given a Fibonacci number m , does there exist a phylogenetic network that displays exactly m distinct phylogenetic trees?” One would then ask “Given a positive integer p , does there exist a phylogenetic network that displays exactly p distinct phylogenetic trees?” That question is the subject of this chapter, and it turns out that, given a positive integer p , there does exist a phylogenetic network that displays exactly p distinct phylogenetic trees. In order to display exactly p trees, where $p \in \mathbb{N}$, we first construct the *caterpillar* on $p + 1$ leaves. A caterpillar on x leaves, where $x \in \mathbb{N}$ and $x \geq 2$, is a rooted directed tree such that the root has a directed edge going into a leaf, call it leaf 1, and has another directed edge going into a vertex α . If $x = 2$ then α is a leaf, call it leaf 2. If $x \geq 3$ then α has a directed edge going into a leaf, call it leaf 2, and has another directed edge going into a vertex β . If $x = 3$ then β is a leaf, call it leaf 3. In this way we have the caterpillar on x leaves.

2.2 The Construction

Theorem 2.1. *Given a positive integer p , there exists a phylogenetic network that displays exactly p distinct phylogenetic trees.*

Proof. First we construct the caterpillar on $p + 1$ leaves. We then set about constructing a network with $p - 1$ reticulation vertices. In order to do this, we first turn each leaf n , except for leaf p and

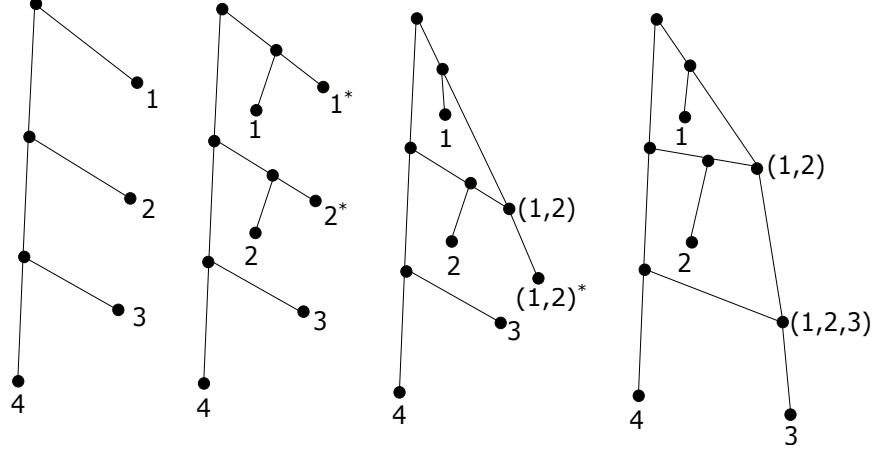


Figure 2.2.1: Here is an example showing a caterpillar with four leaves being converted into a network with two reticulation vertices that displays exactly three trees.

leaf $p + 1$, into a vertex that has two directed edges going into two leaves, call those leaves leaf n and leaf n^* . Next we join leaf 1^* to leaf 2^* and call the resulting vertex $(1, 2)$. The vertex $(1, 2)$ has a single directed edge going into a leaf, call it leaf $(1, 2)^*$. After that we join leaf $(1, 2)^*$ to leaf 3^* and call the resulting vertex $(1, 2, 3)$. The vertex $(1, 2, 3)$ has a single directed edge going into a leaf, call it leaf $(1, 2, 3)^*$. We proceed like this until we get reticulation vertex $(1, 2, \dots, p - 1)$, which has a single directed edge going into leaf $(1, 2, \dots, p - 1)^*$. Join leaf $(1, 2, \dots, p - 1)^*$ to leaf p , call the resulting vertex $(1, 2, \dots, p - 1, p)$. The reticulation vertex $(1, 2, \dots, p - 1, p)$ has a single directed edge going into a leaf, call it leaf p . Now we have a network with $p - 1$ reticulation vertices and $p - 2$ of those reticulation vertices are avoidable reticulation vertices.

Given the above constructed network with $p - 1$ reticulation vertices, we now set out to prove that the network displays exactly p trees. We know that the network can display at most 2^{p-1} distinct trees. Starting at the reticulation vertex called $(1, 2, \dots, p - 1, p)$, we delete the reticulation edge that joins to reticulation vertex $(1, 2, \dots, p - 1)$. As a result of deleting that reticulation edge, every other reticulation vertex is subsequently deleted, which means that the tree that has leaf p and leaf $p + 1$ as a cherry is displayed 2^{p-2} times. Returning to the original network, we delete the other reticulation edge, so that leaf p is now adjacent to reticulation vertex $(1, 2, \dots, p - 1)$. We delete the reticulation edge that joins reticulation vertex $(1, 2, \dots, p - 2)$ to reticulation vertex $(1, 2, \dots, p - 1)$, in this modified network. This results in every other reticulation vertex being deleted, so that the

tree that has leaf p and leaf $p - 1$ as a cherry is displayed 2^{p-3} times. We proceed like this until we get leaf p adjacent to reticulation vertex $(1, 2)$. We see that the tree that has leaf p and leaf 2 as a cherry is displayed exactly once, and likewise the tree that has leaf p and leaf 1 as a cherry is also displayed exactly once. All in all we have p trees, and $2^{p-2} + 2^{p-3} + \dots + 2^1 + 1 + 1 = 2^{p-1}$. Therefore the network displays exactly p trees. \square

Chapter 3

Answering the Driving Question of this Thesis for a Particular Class of Networks

3.1 Introduction

In this chapter, we investigate the driving question of this thesis for a particular class of phylogenetic networks. The driving question of this thesis is: “Given a phylogenetic network N , does there exist a phylogenetic tree with the same leaf set as N that is displayed more than once by N ?” If such a tree exists, then there are two distinct sets of edges in N that yield the same tree. As we know, if N is binary and has k reticulations, then the maximum number of possible trees displayed by N is 2^k . While it was shown independently that the upper bound of 2^k is sharp for so-called “normal networks” in [14, Theorem 1] and [23, Corollary 3.4], little is known about the properties of a phylogenetic network that guarantees that it displays the maximum number of trees, which, of course, is the same as saying that a network never displays a tree twice. Here, we present the first such characterisation for a class of networks that lies strictly between tree-child and tree-sibling networks.

Tree-child and tree-sibling networks are two prominent types of networks arising in the literature. Let N be a network on X . A vertex v of N has the *tree-path property* if there exists a leaf ℓ such that there is a tree-path P from v to ℓ . If such a path exists, then each edge of P is a tree edge and P is the unique directed path from v to ℓ in N . For example, except for the parent common to v and v' , each vertex of the network shown on the left-hand side in Figure 3.1.1 has the tree-path property. We say that N is *tree-child* (e.g. see [24]) if each vertex of N has the tree-path property.

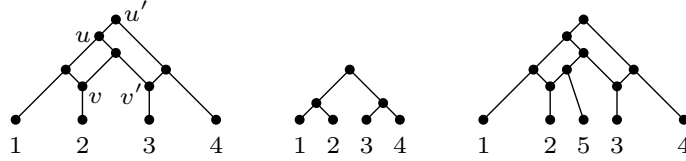


Figure 3.1.1: Left: A phylogenetic network N that displays three trees. Middle: A phylogenetic tree that is displayed twice by N . Right: A phylogenetic network N' that displays four trees. While N and N' are both tree-sibling, only N' also satisfies the stronger tree-child condition.

Equivalently, N is tree-child if each non-leaf vertex u of N has a child v such that v is a tree vertex. Biologically, such networks guarantee that all species that arise from a speciation event (represented by a tree vertex) or a reticulation event exist for a certain period of time before evolving any further. Furthermore, N is *tree-sibling* (e.g. see [25]) if each reticulation has a sibling that is a tree vertex. For example, the network shown on the left-hand side of Figure 3.1.1 is tree-sibling but not tree-child, while the network shown on the right-hand side of the same figure is tree-child (and, hence, also tree-sibling). Observe that, for a fixed set X , the class of tree-child networks on X is a proper subclass of tree-sibling networks on X .

The class of networks on X that is nested strictly between those two classes is the class which has the property that, for each reticulation, at least one of its parents has the tree-path property. The characterisation of when a network in the aforementioned class of networks displays a tree twice is based on a certain type of underlying cycle in a network that will be formally introduced in Section 3.2. Moreover, we will show that such cycles are recognizable in quadratic time, leading to the following theorem.

Theorem 3.1. *Let N be a rooted binary phylogenetic network with leaf set X and suppose that, for each reticulation of N , at least one of its parents is connected to a leaf of N via a directed path that does not contain a reticulation. Then it takes time quadratic in the size of $|X|$ to decide whether or not N displays a rooted phylogenetic tree with leaf set X twice.*

It is worth pointing out that for a network N with the property described in Theorem 3.1, the number of leaves in N does not bound the total number of vertices in N . Hence, for a fixed set X , the class of networks with leaf set X that we consider in this chapter contains infinitely many networks (for example, see Figure 3.1.2, where the directed path from the root of the network to the leaf labeled 1 can be arbitrarily long). In contrast, for a fixed set X , the number of tree-child networks with leaf set X is finite [26].

The remainder of the chapter is organized as follows: In Section 3.2, we introduce the concept

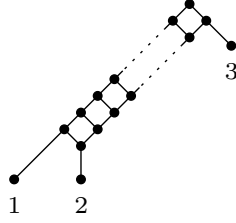


Figure 3.1.2: A phylogenetic network for which each reticulation has a parent connected to the leaf labeled 1 via a directed path that does not contain a reticulation.

of avoidable cycles. We also derive several lemmas in this section that are important in establishing the above-mentioned characterisation, which is presented in Section 3.3. In Section 3.4, we establish Theorem 3.1. The last section contains a remark on tree-child and normal networks.

3.2 Avoidable Cycles

In this section, we describe a certain type of cycle and establish several lemmas that play a role in the characterisation of the next section.

Avoidable cycles. We now extend the concept of avoidability to cycles of a network. Let N be a network on X , and let v be a reticulation of N . Let u be a tree vertex of N such that there exists two directed paths P_1 and P_2 from u to v whose vertex sets, apart from u and v , are disjoint. We call the underlying cycle induced by the union of the vertex sets of P_1 and P_2 a *2-path cycle* of N , where u is the *source* vertex and v is the *sink* vertex. It is easily seen that each reticulation of N is the sink of at least one 2-path cycle in N .

Let C be a 2-path cycle of N with source u and sink v . Let H be a subset of the vertex set of C such that, for each leaf $\ell \in X$, at least one of the following holds:

- (i) there is a directed path from the root of N to ℓ which avoids every vertex in C , or
- (ii) there is a directed path from the root of N to ℓ for which the last vertex in the path meeting C is contained in H .

We refer to H as a *hitting set* of C . Furthermore, H is *minimum* if C has no hitting set H' with $|H'| < |H|$. If there exists a hitting set of C with at most two elements, we say that C is *avoidable*. A simplified phylogenetic network that has an avoidable cycle and summarizes the basic idea of such a cycle is shown in Figure 3.2.1. Moreover, for a more explicit example, the network shown on the left-hand side of Figure 3.1.1 has a 2-path cycle C with source u and sink v that is avoidable,

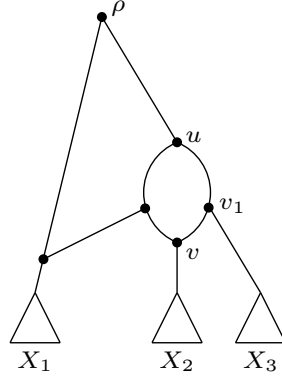


Figure 3.2.1: A 2-path cycle C of a network N on $X = X_1 \cup X_2 \cup X_3$ with source u and sink v . Note that $\{v, v_1\}$ is a hitting set of C because X can be partitioned into three sets X_1 , X_2 , and X_3 such that, for each $\ell_1 \in X_1$, there exists a directed path from ρ to ℓ_1 that avoids every vertex in C , and, for each $\ell_2 \in X_2$ (resp. $\ell_3 \in X_3$), there exists a directed path from ρ to ℓ_2 (resp. ℓ_3) for which the last vertex on that path that meets a vertex in C is v (resp. v_1). Thus C is an avoidable cycle. Except for the edge joining v_1 and v , lines indicate directed paths in N . Furthermore, the three triangles indicate subnetworks of N . While omitted for the sake of simplicity, these subnetworks as well as C may be further interwoven among themselves and among each other.

and a 2-path cycle with source u' and sink v' that is unavoidable. Note that C is avoidable because there exists directed paths from the root of the network to leaves 3 and 4 that do not meet C .

The next lemma gives a sufficient, but not a necessary, condition for guaranteeing that a network displays a tree twice.

Lemma 3.1. *Let N be a network on X , and let v be a reticulation of N . If v is the sink of an avoidable cycle, then N displays a tree on X twice.*

Proof. Suppose that v is the sink of an avoidable cycle C . Then there is a hitting set H of C such that $|H| \leq 2$. Furthermore, for each $\ell \in X$, there is a directed path P_ℓ in N from the root to ℓ such that either P_ℓ avoids every vertex of C or the last vertex of P_ℓ meeting C is an element of H .

Now, let T be a tree on X displayed by N whose edge set, up to degree-2 vertices, is a subset of $\bigcup_{\ell \in X} P_\ell$. Since H contains at most two elements, T has a subtree that can be detached by deleting a single edge and whose leaf set contains precisely each element $\ell \in X$ for which the last vertex of P_ℓ meeting C is an element of H . Let e_1 and e_2 denote the reticulation edges incident with v , and let S be a switching of N that yields T . By construction, it is now easily seen that the two switchings $(S - \{e_1, e_2\}) \cup \{e_1\}$ and $(S - \{e_1, e_2\}) \cup \{e_2\}$ both yield T . Hence N displays a tree on X twice. \square

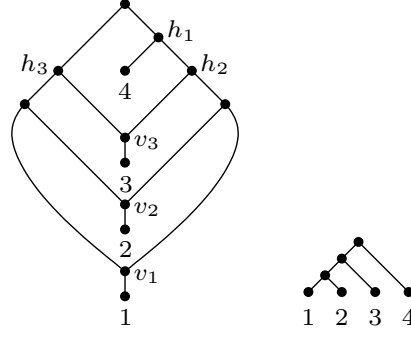


Figure 3.2.2: A phylogenetic network (left) that displays the tree shown on the right twice. Moreover, N has no avoidable cycle because each 2-path cycle of N with sink v_i , for $i \in \{1, 2, 3\}$, has a minimum hitting set of size at least three. For example, $\{h_1, h_2, v_3\}$ and $\{h_1, h_3, v_3\}$ are the two unique minimum hitting sets of the 2-path cycle of N with sink v_3 .

The converse of Lemma 3.1 does not hold. For example, Figure 3.2.2 shows a network that has no avoidable cycle, but displays a tree twice.

We end this section with a concept and an lemma that is used in the rest of the chapter. Let N be a network, and let v be a reticulation of N . A parent of v is a *distinguished parent* if it has the tree-path property and, if both parents of v have the tree-path property, then it is not an ancestor of the other parent. Note that, if v has a parent that has the tree-path property, then v has at least one distinguished parent. Moreover, if v has two distinguished parents, then v is not the sink of an avoidable cycle in N . Referring back to Figure 3.1.1, each of the two reticulations in the network shown on the left has exactly one distinguished parent, while each of the two reticulations in the network shown on the right of the same figure has two distinguished parents.

The following lemma immediately follows from the definition of an avoidable cycle and recalling that such a cycle has a hitting set of size at most two.

Lemma 3.2. *Let N be a network with no avoidable reticulation, and let v be a reticulation of N . If v has a distinguished parent, say v_1 , and v is the sink of an avoidable cycle C in N , then $\{v_1, v\}$ is the unique minimum hitting set of C .*

3.3 Characterisation

In this section, we characterise when a network with at least one parent of each reticulation having the tree-path property displays a tree twice. This characterisation is in terms of avoidable reticulations and avoidable cycles. We will see in the next section that this result leads naturally to a quadratic-time algorithm that decides whether or not such a network displays a tree twice.

We start by describing an operation that involves a deletion of a reticulation in a network. Let N be a network with no avoidable reticulation and, for each reticulation, at least one of its parents has the tree-path property. Let ρ be the root of N , and let v be a reticulation of N whose strict descendants are all tree vertices. Since N is acyclic such a reticulation exists. Obtain a rooted acyclic digraph N' from N by deleting v and contracting any resulting vertex of in-degree one and out-degree one. Such vertices correspond to v_1 and v_2 and, provided neither is ρ , there are two contractions. If v_1 or v_2 is ρ , then delete ρ as well. We say that N' is obtained from N by a *reticulation deletion* relative to v . The next lemma shows that N' preserves the two properties of N that distinguish it.

Note 1. We use D_u to denote the subset of X whose elements are precisely the descendants of u .

Lemma 3.3. *Let N be a network on X with no avoidable reticulation. Suppose that N has the tree-path property for at least one parent of each reticulation. Let N' be the rooted acyclic digraph obtained from N by a reticulation deletion relative to a reticulation v . Then N' is a network on $X - D_v$ with no avoidable reticulation and, for each reticulation, at least one of its parents has the tree-path property.*

Proof. Let ρ denote the root of N . Furthermore, let v_1 and v_2 denote the parents of v . Without loss of generality, we may assume that v_1 is a distinguished parent of v . Let m denote a leaf in N with the property that there is a tree-path from v_1 to m . Now, since each reticulation in N is unavoidable, v_1 and v_2 are tree vertices. Using this fact, as well as the property that at least one parent of each reticulation has the tree-path property in N , it is easily checked that N' is indeed a phylogenetic network on $X - D_v$ (with no parallel edges).

We next show that each reticulation in N' is unavoidable, and at least one parent of each reticulation in N' has the tree-path property. The latter certainly holds as no such tree-path in N contains either (v_1, v) or (v_2, v) . Now, let w be a reticulation in N' . If w is avoidable in N' , then, as w is unavoidable in N , there is a leaf $\ell \in D_v$ such that every directed path in N from ρ to ℓ meets w . Moreover, there is a directed path P_m from ρ to m in N avoiding w . Since P_m extends the unique tree-path from v_1 to m , it follows that, by making use of the first part of P_m from ρ to v_1 , we can construct a directed path from ρ to ℓ that uses the edge (v_1, v) and avoids w in N ; a contradiction. Thus each reticulation in N' is unavoidable. \square

The next theorem is the aforementioned characterisation.

Theorem 3.2. *Let N be a network on X . Suppose that at least one parent of each reticulation in N has the tree-path property. Then N displays a tree on X twice if and only if N contains an*

avoidable reticulation or an avoidable cycle.

Proof. Let ρ denote the root of N . If N contains an avoidable reticulation or an avoidable cycle, then, by Lemmas 1.3 and 3.1, N displays a tree on X twice.

Now, suppose that N contains neither an avoidable reticulation nor an avoidable cycle. Let k be the number of reticulations in N . We will show by induction on k that N does not display a tree on X twice. If $k = 0$, then N is a tree on X and the result holds. Now assume that $k \geq 1$ and that the result holds for all networks with $k - 1$ reticulations. Let v be a reticulation of N whose strict descendants are all tree vertices, and let v_1 and v_2 be the two parents of v . Without loss of generality, assume that v_1 is a distinguished parent of v . Furthermore, let m denote a leaf in N with the property that there is a tree-path from v_1 to m . Let N' be the rooted acyclic digraph obtained from N by applying a reticulation deletion relative to v . It follows, by Lemma 3.3, that N' is a network on $X - D_v$ with no avoidable reticulation and, for each reticulation, at least one parent has the tree-path property.

To apply the induction assumption, we next show that N' contains no avoidable cycles. Suppose to the contrary that N' has an avoidable cycle C' with sink t . Let t_1 and t_2 denote the parents of t and, without loss of generality, assume that t_1 is a distinguished parent of t . By Lemma 3.2, it follows that $\{t_1, t\}$ is the unique minimum hitting set H' of C' . Let C denote the 2-path cycle in N induced by C' in N' . Since each tree vertex in N and N' has out-degree exactly 2, and t_1 has the tree-path property in N' , it follows that t_1 is not contained in $\{v_1, v_2\}$, so (t_1, t) is an edge in C' and C . Now, let P'_m be a directed path from the root of N' to m such that either P'_m avoids C' or the last vertex of P'_m that meets C' is contained in H' . As C' is an avoidable cycle in N' , such a path exists. Now, if $v_2 = \rho$ and (v_2, v_1) is an edge in N , let v_p denote the child of v_1 in N such that $v_p \neq v$; otherwise, let v_p denote the parent of v_1 in N . Note that the unique directed path from v_p to m in N' is a subpath of P'_m .

We next consider two cases. First, assume that the subpath of P'_m in N' from v_p to m either avoids every vertex in C' or $v_p \in \{t_1, t\}$. By the existence of P'_m in N' , we have that, for each leaf $\ell \in D_v$, there exists a directed path P_ℓ from ρ to ℓ in N that uses the edge (v_1, v) such that P_ℓ avoids every vertex of C or the last vertex of P_ℓ that meets C is contained in $\{t_1, t\}$. Furthermore, as (t_1, t) is an edge in C , we have that H' is a hitting set of C in N . In particular, as C' is an avoidable cycle in N' , it follows that C is an avoidable cycle in N ; a contradiction.

Second, assume that the subpath of P'_m from v_p to m in N' does not avoid every vertex in C' and $v_p \notin \{t_1, t\}$. As C is unavoidable in N , v_1 is either a vertex of C or the source of C is a strict descendant of v_1 . In the latter case, it is easily checked that, as C' is avoidable in N' , C is avoidable

in N ; a contradiction. We may therefore assume that v_1 is a vertex of C . If there is an element $\ell \in D_v$ for which there is a directed path in N from ρ to ℓ through v_2 such that either it avoids C , or it meets C and the last vertex it meets in C is t or t_1 , then all elements in D_v have such a path. In turn, this implies that C is avoidable in N ; a contradiction. Hence, for all $\ell \in D_v$, every directed path from ρ to ℓ through v_2 meets a vertex of C and the last such vertex is neither t nor t_1 . Let r denote such a vertex of C , and let P_r denote a directed path from r to v_2 in N . We may assume that r is the only vertex of P_r meeting C . Potentially, P_r may consist of the single vertex v_2 . Now, let D be the unique 2-path cycle in N with sink v whose vertex set is the union of $V(P_r) \cup \{v\}$ and a subset of the vertices in C , and whose edge set is $E(P_r) \cup \{(v_1, v), (v_2, v)\}$ a subset of the edges in C , where $V(P_r)$ and $E(P_r)$ are the vertex and edge sets of P_r , respectively. Let X_{v_1} denote the subset of X such that $p \in X_{v_1}$ precisely if $p \in D_v$ or there is a path from v_1 to p that avoids D except for v_1 . Since v is not the sink of an avoidable cycle in N , the set $X - X_{v_1}$ is non-empty. In particular, there exists a leaf $q \in X - X_{v_1}$ with the property that every directed path from ρ to q in N meets D and the last vertex meeting D is neither v nor v_1 . Moreover, since C' is avoidable in N' , at least one such path, say P_q , does not meet a vertex of C in N or the last vertex meeting C in N is an element in $\{t_1, t\}$. If the last vertex of P_q that meets C in N is either t_1 or t , it is easily checked that there is a path from ρ to q such that the last vertex on this path meeting D is v_1 ; a contradiction. We may therefore assume that P_q does not meet a vertex of C . Hence, $V(P_r) - \{r\}$ is non-empty and, in particular, P_q meets D in a vertex of $V(P_r) - \{r\}$. But then there is a directed path in N from ρ to ℓ using P_q that avoids every vertex in C , in which case, C is avoidable in N ; a contradiction.

We now proceed with the induction. Since N' has $k - 1$ reticulations, it follows by the induction assumption that N' does not display a tree on $X - D_v$ twice. Let T' be a tree on $X - D_v$ that is displayed by N' , and let S' be a switching that yields T' . Now consider the two switchings $S_1 = S' \cup \{e_1\}$ and $S_2 = S' \cup \{e_2\}$, where $e_1 = (v_1, v)$ and $e_2 = (v_2, v)$. For completeness, if S' contains an edge (w_1, w) , where w_1 is the parent of v_2 and w is a child of v_2 in N , then replace (w_1, w) with (v_2, w) in S_1 and S_2 . Let C be a 2-path cycle in N whose sink is v . It is easily checked that C exists. Furthermore, let ℓ be an element in D_v , and let q be an element in X such that the last vertex of each directed path from ρ to q in N that meets C is neither v nor v_1 . As C is not avoidable, such a q exists. Then S_1 yields a tree \mathcal{T}_1 on X that contains the triple $\ell m|q$ while S_2 yields a tree \mathcal{T}_2 on X that contains the triple $\ell q|m$ or $qm|\ell$ and, thus, $\mathcal{T}_1 \not\cong \mathcal{T}_2$, by Lemma 1.2. Applying this argument to each of the trees on $X - D_v$ displayed by N' , it follows that N does not display a tree on X twice; thereby completing the proof of the theorem. \square

3.4 Quadratic-Time Algorithm

Making use of the characterisation Theorem 3.2, in this section we establish Theorem 3.1. If N is a network with n vertices, then, as each vertex of N has degree at most three, the number of edges in N is at most $\frac{3}{2}n$. We will implicitly use this fact throughout the section.

We start by showing that the total number of vertices in a certain type of network N on X is bounded by a function that is linear in the size of X . Eventually, this will enable us to get the overall running time to be quadratic in $|X|$.

Lemma 3.4. *Let N be a network on X with no avoidable reticulation, and suppose that N has the tree-path property for at least one parent of each reticulation. Let k be the number of reticulations in N , and let n be the total number of vertices in N . Then $k \leq |X|$ and, in particular, $n < 4|X|$.*

Proof. If $k = 0$, then the result clearly holds. So assume that the result holds for all networks with fewer than k reticulations. Let N' be a network obtained from N by applying a reticulation deletion relative to a reticulation v in N . It follows, by Lemma 3.3, that N' is a network on $X - D_v$ with no avoidable reticulation and, for each reticulation, at least one parent has the tree-path property. Moreover, N' has $k - 1$ reticulations and at most $|X| - 1$ leaves. Therefore, by induction,

$$k - 1 \leq |X - D_v| \leq |X| - 1,$$

and so $k \leq |X|$. To establish the second part, we use a result from [26, Equation 5] whose authors have shown that $|X| + k = \frac{n+1}{2}$. Since $k \leq |X|$, it follows that

$$n = 2(|X| + k) - 1 \leq 4|X| - 1 < 4|X|,$$

thereby establishing the second inequality of the lemma. \square

Corollary 3.1. *Let N be a network on X that has the tree-path property for at least one parent of each reticulation. If N has at least $4|X|$ vertices, then N displays a tree on X twice.*

Proof. It follows by the contrapositive of Lemma 3.4 that N has an avoidable reticulation. Hence, by Lemma 1.3, N displays a tree on X twice. \square

Following on from Corollary 3.1, the next lemma shows that we can decide quickly if a network on X has at least $4|X|$ vertices.

Lemma 3.5. *Let N be a network on X . It takes time linear in $|X|$ to decide if N has at least $4|X|$ vertices.*

Proof. The result follows by applying a breadth-first search traversal to N that keeps track of the number of previously visited distinct vertices in N and either returns the number n of vertices in N if $n < 4|X|$ or stops if $4|X|$ distinct vertices have been traversed. Since the running time of a breadth-first search algorithm applied to N is $O(\frac{3}{2}n + n)$ [27], the lemma now follows. \square

We next establish a lemma on avoidable cycles and then state an algorithm that recognizes whether or not a reticulation is the sink of an avoidable cycle in a network with no avoidable reticulations and, for each reticulation, at least one parent has the tree-path property.

Lemma 3.6. *Let N be a network with no avoidable reticulation, and suppose that at least one parent of each reticulation in N has the tree-path property. Let v be a reticulation in N with parents v_1 and v_2 say, where v_1 is a distinguished parent of v . If v is the sink of an avoidable cycle C , and P_1 and P_2 are the two directed paths whose union is C with v_i lying on P_i , then, apart from v , the path P_1 contains at most one reticulation and the path P_2 contains no reticulations. Moreover, C is the unique avoidable cycle with sink v .*

Proof. Let ρ denote the root of N . It follows, by Lemma 3.2, that $\{v, v_1\}$ is the unique hitting set of C . We first show that P_2 contains no reticulations except for v . Assume that w is a reticulation lying on P_2 such that $w \neq v$. Amongst all such reticulations, choose w so that the only reticulation in P_2 after w is v . Since w is unavoidable, there exists a leaf q such that every directed path from ρ to q contains w . In particular, there exists a directed path from ρ to q , say P_q , such that, as C is avoidable, the last vertex of P_q meeting C is either v or v_1 . But then, as w is not the source of C , there is a directed path from ρ to q using P_1 that avoids w ; a contradiction. Thus P_2 contains no reticulations except v .

We next show that P_1 contains at most one reticulation except for v . Assume that w is a reticulation lying on P_1 such that $w \neq v$. Like above, choose w so that amongst all such reticulations the only reticulation after w in P_1 is v . Let w_1 and w_2 be the parents of w in N . Without loss of generality, we may assume that w_1 is a distinguished parent of w . Since w_1 has the tree-path property, there is a leaf q with the property that there is a tree-path from w_1 to q . Since C is avoidable and $\{v, v_1\}$ is the unique hitting set of C , it follows that w_1 does not lie on P_1 ; otherwise, a hitting set of C has size at least three. Thus w_2 lies on P_1 . Now assume that P_1 contains a reticulation t other than v and w . Choose t so that the only reticulations after t in P_1 are w and v . Since t is unavoidable, there exists a leaf r such that every directed path from ρ to r contains t . Moreover, as C is avoidable, there exists at least one such path, say P_r , such that the last vertex of P_r meeting C is either v or v_1 . Now, let P_q be a directed path from ρ to q and observe that P_q

contains as a subpath the tree-path from w_1 to q . Since N is acyclic and C is avoidable, P_q does not meet C . But then there is a directed path from ρ to r using P_q to w_1 , the unique path from w_1 to v_1 , and the subpath of P_r from v_1 to r . In particular, this path avoids t ; a contradiction. Hence, P_1 contains at most one reticulation other than v .

To see that C is the unique avoidable cycle with sink v in N , first note that P_2 contains no reticulations except v . Furthermore, P_1 contains at most one reticulation (other than v) and, if it contains such a reticulation w , then P_1 has no choice with regards to which parent of w it meets. Since no 2-path cycle of N with sink v that contains v , v_1 , and a parent of w that has the tree-path property is avoidable, the uniqueness of C now follows. \square

The previous lemma provides insights into how to decide whether or not a reticulation is the sink of an avoidable cycle in a network N on X with no avoidable reticulation and for which the tree-path property holds for at least one parent of each reticulation. We next summarize these insights in the form of an algorithm, called `AVOIDABLECYCLE`. Subsequently, we will establish that `AVOIDABLECYCLE` works correctly and that its running time is linear in the size of X .

Algorithm: `AVOIDABLECYCLE`

Input: A network N on X with no avoidable reticulation and, for each reticulation, at least one parent has the tree-path property. A reticulation v of N with parents v_1 and v_2 say, where v_1 is a distinguished parent of v .

Output: Return “yes” if v is the sink of an avoidable cycle in N ; otherwise, return “no.”

Step 1 Set $P_2 = u_1, u_2, \dots, u_l$ to be the (unique) maximal directed path in N with $u_{l-1} = v_2$ and $u_l = v$ such that, except for v , each vertex on P_2 is a tree vertex.

Step 2 Set $P_1 = w_1, w_2, \dots, w_m$ to be the (unique) maximal directed path in N with $w_{m-1} = v_1$ and $w_m = v$ such that the following three properties are satisfied: (i) w_1 is a tree vertex, (ii) P_1 contains at most one reticulation other than v , and (iii) except for v_1 and, possibly v_2 , no vertex on P_1 that is a parent of a reticulation in N , has the tree-path property.

Step 3 If P_1 and P_2 have no common tree vertex, then return “no.” Otherwise, let C be the 2-path cycle of N induced by subpaths of P_1 and P_2 with source u and sink v , where u is the last tree vertex in P_1 and P_2 common to both paths.

Step 4 Let X' be the subset of X such that $\ell \in X'$ if and only if there is a directed path from either v_1 or v to ℓ avoiding all other vertices of C .

Step 5 For each leaf q in $X - X'$, check whether there is a directed path from the root of N to q avoiding all vertices of C . Return “yes” if there exists such a path for all q ; otherwise, return “no.”

Lemma 3.7. *Let N be a network on X with no avoidable reticulation. Suppose that at least one parent of each reticulation in N has the tree-path property. Let v be a reticulation in N . Calling `AVOIDABLECYCLE` for N and v returns “yes” if and only if v is the sink of an avoidable cycle. Furthermore, the running time of `AVOIDABLECYCLE` in this call is linear in the number of vertices in N .*

Proof. Let ρ denote the root of N , and let v_1 and v_2 denote the parents of v . Without loss of generality, we may assume that v_1 is a distinguished parent of v . Furthermore, let n denote the number of vertices in N . Throughout the proof, we use the same notation as in the description of `AVOIDABLECYCLE`.

We first show that `AVOIDABLECYCLE` works correctly. Suppose that C' is an avoidable cycle of N with sink v . Then, by Lemma 3.6, C' is unique. Applying `AVOIDABLECYCLE` to N and v , it follows, by Lemma 3.6 and the construction described in `AVOIDABLECYCLE`, that C' is the 2-path cycle C constructed in Step 3 of the algorithm. By the definition of an avoidable cycle, Step 5 returns “yes.” Now suppose that N has no avoidable cycle with sink v . Applying `AVOIDABLECYCLE` to N and v , there are two cases to consider depending on whether or not P_1 and P_2 meet in Step 3. If P_1 and P_2 do not meet at a tree vertex, then Step 3 returns “no.” Therefore, assume that P_1 and P_2 do meet at a tree vertex. Then, as v is not the sink of an avoidable cycle in N , there is some leaf $q \in X - X'$ such that every path from ρ to q meets C , in which case Step 5 returns “no.” Hence, `AVOIDABLECYCLE` correctly determines if v is the sink of an avoidable cycle in N .

We now turn to the running time of `AVOIDABLECYCLE`. Starting at v_2 and traversing edges in the opposite direction to determine P_2 takes time linear in n . Similarly, determining P_1 takes time linear in n . However, if P_1 contains a reticulation v' , distinct from v , then one has additionally to determine which of its two parents, say v'_1 and v'_2 , have the tree-path property. A naive way to do this is the following. Let $(r_1, r_2, \dots, r_{|X|})$ be an ordering on the leaves of N . In turn, for each r_i , let P_{r_i} be the unique maximal directed path in N that ends in r_i such that each vertex on P_{r_i} is a tree vertex and, except for the first vertex of P_{r_i} , no vertex is contained in a path P_{r_j} with $1 \leq j < i \leq |X|$. If there exists an r_i such that P_{r_i} meets v'_k with $k \in \{1, 2\}$, then v'_k has the

tree-path property. Collectively, this takes time linear in n . Clearly, Step 3 can be done in time linear in n and, so, it remains to check the running time of Steps 4 and 5. For Step 4, delete the vertices in C that are neither v nor v_1 , and then determine, for each leaf ℓ , if there is a directed path from v_1 to ℓ in the resulting directed graph, in which case, $\ell \in X'$. Here we can, for example, use a depth-first search traversal [27] starting at v_1 and, so, this step takes time linear in n . An analogous approach can be done for Step 5. We conclude that the running time of `AVOIDABLECYCLE` is linear in n . \square

We are now in a position to prove Theorem 3.1.

Theorem 3.1. *Let N be a network on X and suppose that N has the tree-path property for at least one parent of each reticulation. It takes time quadratic in the size of X to decide if N displays a tree on X twice.*

Proof. First, by Lemma 3.5, we can decide in time linear in $|X|$ if N has at least $4|X|$ vertices. If N has at least that many vertices, then, by Corollary 3.1, N displays a tree on X twice. We may therefore assume that N has at most $4|X|$ vertices.

We complete the proof by showing that it takes time quadratic in $|X|$, to decide whether or not N has an avoidable reticulation or an avoidable cycle which is, by Theorem 3.2, a necessary and sufficient condition for N to display a tree on X twice. Let v be a reticulation in N . Deciding if v is avoidable is easily checked in time that is linear in the size of N , which is at most $4|X|$. For example, one way is to simply delete v from N and then use a depth-first search [27], whose running time is linear in $|X|$, to decide whether there is a directed path from the root to each vertex in X in the resulting directed graph. Since the number of reticulations in N is at most $|X|$ (see Lemma 3.4), deciding whether or not N has an avoidable reticulation takes time quadratic in $|X|$. Now we may assume that N has no avoidable reticulation. It then follows, by Lemma 3.7, that it takes time linear in the number of vertices in N and, hence, by Lemma 3.5, time linear in $|X|$, to decide if v is the sink of an avoidable cycle in N using `AVOIDABLECYCLE`. Applying this algorithm to each reticulation in N to decide if there exists a reticulation that is the sink of an avoidable cycle takes time quadratic in $|X|$. The theorem now follows. \square

3.5 Remark on Tree-Child and Normal Networks

As tree-child networks are a subclass of the networks in which each reticulation has at least one parent that satisfies the tree-path property, it immediately follows, by Theorem 3.1, that it can be decided quickly whether or not a tree-child network displays a tree twice. Curiously, since each

vertex of a tree-child network N has the tree-path property, it is tempting to assume that N never displays a tree twice and therefore has no avoidable cycles. However, this is not necessarily true. To see this, consider a reticulation v of N and its two parents v_1 and v_2 . If v_1 has the tree-path property and v_2 is an ancestor of v_1 , then it is possible for v to be contained in an avoidable cycle. In [28], Willson refers to a tree-child network that does not have a reticulation for which one parent is an ancestor of the other parent as a *normal network*. Noting that a normal network does not have an avoidable cycle as every 2-path cycle has a minimum hitting set of size at least three, the next corollary is now an immediate result of Theorem 3.2.

Corollary 3.2. *Let N be a normal network on X . Then N does not display a tree on X twice.*

Chapter 4

Characterising when a Phylogenetic Network Displays a Tree Twice in Terms of Substructures in the Network

4.1 Introduction

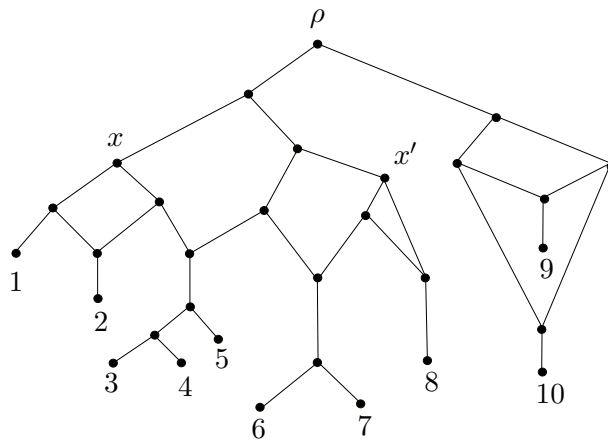
This chapter is about characterising when an arbitrary phylogenetic network displays a tree twice in terms of local substructures inherent in the network. These local substructures arise due to the presence of reticulation vertices, and the local substructures, when isolated, are themselves phylogenetic networks. In Section 4.2 we identify the local substructures inherent in a network and show that the local substructures are themselves networks. In Section 4.3 we show how the local substructures can be represented by a tree, so that essentially the local substructures are tree-like. Section 4.4 presents a proposition that is a crucial part of the algorithm that decides whether or not a network displays a tree twice, as well as describing what it means to reduce a representation. Section 4.5 shows that a network, displayed by a network without any avoidable reticulation vertices, is free of avoidable reticulation vertices. In Section 4.6 we present the characterisation of when a network displays a tree twice in terms of the local substructures identified in Section 4.2.

4.2 Basic Networks and Basic Cyclic Pairs

The networks for which it is easy to decide whether or not they display a tree twice are networks that have only one level of reticulation vertices; this means that the descendants of any reticulation vertex in the network are non-reticulation vertices. We call such a network a basic network. The subnetworks of a basic network are basic networks, so one can not get any simpler than a basic network in terms of substructures. One does find, however, that some basic networks have a more direct way of deciding whether or not they display a tree twice than other basic networks; these networks are called basic cyclic pairs. A basic cyclic pair is an ordered tuple (w, v, P_1, P_2) and a basic network B , where w is a tree vertex, v is a reticulation vertex, and P_1, P_2 are two edge disjoint directed paths connecting w to v such that every directed path from the root to a reticulation vertex in the network passes through w . Since a basic cyclic pair is identified by a reticulation vertex v , the reticulation vertex v is special, so that defining when a basic cyclic pair displays a tree twice involves the reticulation vertex v . We say that a basic cyclic pair displays a tree twice when there exists two switchings such that v 's reticulation edges are split between the two switchings, which means that one of v 's reticulation edges is in one of the switchings whilst the other reticulation edge is in the other switching, and those two switchings yield the same tree.

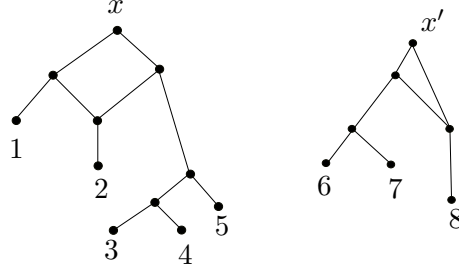
Definition 4.1. A *basic network* is a phylogenetic network N such that for every reticulation vertex v in N there is no other reticulation vertex v' where there is a directed path from v to v' .

Example 4.1. The network shown is a basic network. The network shown in Example 1.5 is not a basic network.



Definition 4.2. Let N be a network on leaf set X . A network N_1 on leaf set $X_1 \subseteq X$ is a *subnetwork* of N when there exists a tree-vertex x in N such that deleting all the vertices and edges that can not be reached from x via a directed path in N then suppressing all vertices with in-degree one and out-degree one results in N_1 .

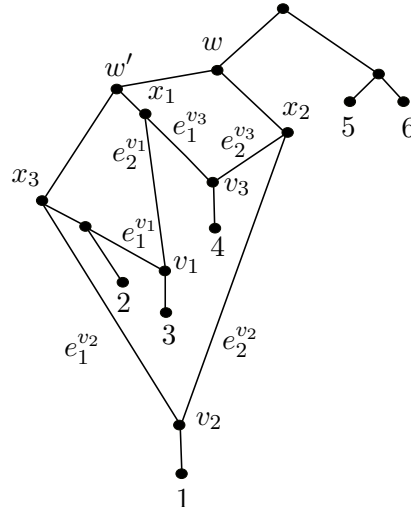
Example 4.2. The network shown on the left was obtained from the network shown in Example 4.1 by deleting all the vertices and edges that can not be reached from x via a directed path in the network shown in Example 4.1 then suppressing all vertices of in-degree one and out-degree one. Similarly the network shown on the right was obtained from the network shown in Example 4.1 with respect to x' .



Definition 4.3. Let B be a basic network, and let w be a tree-vertex in B such that every directed path from the root of B to a reticulation vertex in B passes through w . Let v be a reticulation vertex in B such that there exists two directed paths P_1, P_2 such that $E(P_1) \cap E(P_2) = \emptyset$ and $V(P_1) \cap V(P_2) = \{w, v\}$. The ordered tuple (w, v, P_1, P_2) and a basic network B is called a *basic cyclic pair*. We denote a basic cyclic pair by $B-(w, v, P_1, P_2)$.

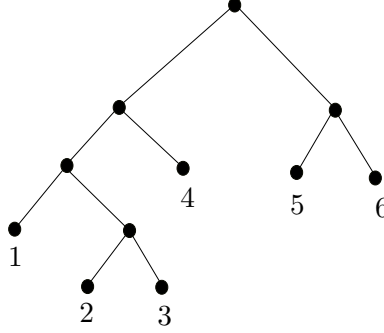
Note 2. This definition allows a parallel pair, as reticulation edges are given distinct labels. In fact, if v is a reticulation vertex then we label v 's reticulation edges e_1^v, e_2^v .

Example 4.3. The basic network shown, called B , has two basic cyclic pairs. One basic cyclic pair is $B-(w, v_3, P_1, P_2)$, where $P_1 = \{(w, w'), (w', x_1), (x_1, v_3)\}$ and $P_2 = \{(w, x_2), (x_2, v_3)\}$. The other basic cyclic pair is $B-(w, v_2, P'_1, P'_2)$, where $P'_1 = \{(w, w'), (w', x_3), (x_3, v_2)\}$ and $P'_2 = \{(w, x_2), (x_2, v_2)\}$. The network shown in Example 4.1 has no basic cyclic pairs.



Definition 4.4. A basic cyclic pair $B-(w, v, P_1, P_2)$ *displays a tree twice* when there exists two switchings S_1, S_2 of B that yield the same tree, and one of v 's reticulation edges is in S_1 whilst the other is in S_2 .

Example 4.4. The basic cyclic pair $B-(w, v_3, P_1, P_2)$ from Example 4.3 displays the tree shown twice via $S_1 = \{e_1^{v_2}, e_1^{v_1}, e_1^{v_3}\}$ and $S_2 = \{e_1^{v_2}, e_1^{v_1}, e_2^{v_3}\}$. The other basic cyclic pair $B-(w, v_2, P'_1, P'_2)$ from Example 4.3 does not display the tree shown twice as $e_1^{v_2} \in S_1$ and $e_2^{v_2} \notin S_2$.



4.3 Representing a Basic Cyclic Pair

When it comes to analysing a basic cyclic pair, we just want the information that is relevant in regards to deciding when the basic cyclic pair displays a tree twice. It is also preferable to work with existing concepts that have well-established notation. Since every directed path from the root to a reticulation vertex passes through w , we have w as the root of the tree representing the basic cyclic pair. First we relabel each leaf n in the leaf set of the basic cyclic pair with ℓ_n . Now for each reticulation vertex v_i , where $i \in A$ and A is a finite subset of the natural numbers, there is a tree structure below it, so delete the edge going out of v_i and delete any components that do not contain w . Now v_i has two directed edges going into it and no edge going out), so we split the vertex v_i into two leaves $v_i^{e_1}$ and $v_i^{e_2}$ where $v_i^{e_1}$ is the finish vertex of the edge $e_1^{v_i}$ and $v_i^{e_2}$ is the finish vertex of the edge $e_2^{v_i}$. Now we have a tree that represents the basic cyclic pair, and the tree contains all the information that is needed in order to decide whether or not the basic cyclic pair that it represents displays a tree twice.

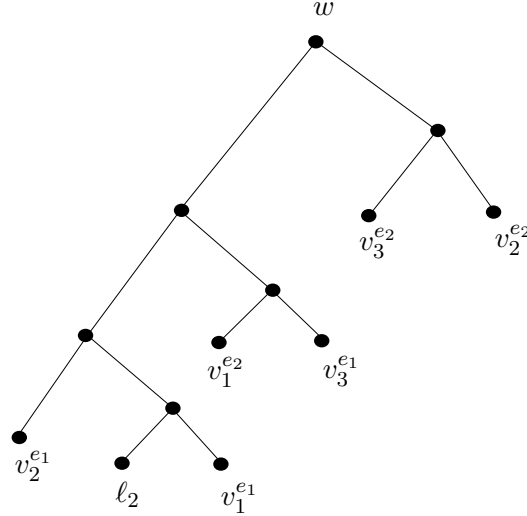
Definition 4.5. The tree \mathcal{T} that represents a basic cyclic pair $B-(w, v, P_1, P_2)$ is obtained from the basic cyclic pair as follows:

1. For each leaf n in the leaf set of B , relabel n with ℓ_n .
2. For each reticulation vertex v_i in B , where $i \in A$ and A is a finite subset of the natural numbers, delete the edge going out of v_i .

3. For each vertex v_i split it into two leaves $v_i^{e_1}$ and $v_i^{e_2}$ such that $v_i^{e_1} = \text{fin}(e_1^{v_i})$ and $v_i^{e_2} = \text{fin}(e_2^{v_i})$, where $\text{fin}(e_1^{v_i})$ means the finish vertex of the directed edge $e_1^{v_i}$.
4. Delete the edge going into w and delete any components that do not contain w .

Note 3. In $B-(w, v, P_1, P_2)$ $v = v_j$ for some $j \in \mathbb{N}$.

Example 4.5. The tree \mathcal{T} shown represents the basic cyclic pair $B-(w, v_3, P_1, P_2)$ from Example 4.3. The tree shown also represents the basic cyclic pair $B-(w, v_2, P'_1, P'_2)$ from Example 4.3.



4.3.1 What does it mean for the representation of a basic cyclic pair to display a tree twice?

The tree that represents a basic cyclic pair has two distinct groups of leaves. One group of leaves represents reticulation edges in the basic cyclic pair whilst the other represents leaves that can be got to from a tree-path starting at w . The first group of leaves can be restricted in the following way: for every leaf $v_i^{e_j}$, where $j \in \{1, 2\}$ and $i \in \mathbb{N}$, exactly one of $v_i^{e_1}, v_i^{e_2}$ is included in the leaf restriction. The second group of leaves is fixed so that any leaf restriction of the tree must include the second group. We see that these special leaf restrictions correspond to a switching of a basic cyclic pair and vice versa, so that a basic cyclic pair displays a tree twice if and only if its tree representation displays a tree twice. There is one more thing that needs to be done for a coherent representation of displaying a tree twice: After a leaf-restriction is applied to the tree representing a basic cyclic pair, all the leaves $v_i^{e_j}$ need to be relabeled ℓ_{v_i} . This ensures that the trees that are displayed by the representation are embedded in the trees displayed by the basic cyclic pair that it represents.

Definition 4.6. Let T be a tree on leaf set X . A *leaf restriction* R of T is a set such that $R \subseteq X$.

Definition 4.7. Let T be a tree on leaf set X , and let X_1 be a leaf restriction of T . The *leaf restriction* X_1 applied to T is obtained from T by deleting all the vertices and edges that are not on a directed path from the root of T to an element of X_1 . Then deleting any vertices with in-degree zero and suppressing any vertices of in-degree one and out-degree one. We denote the leaf restriction X_1 applied to T by $T|X_1$.

Definition 4.8. Let \mathcal{T} be the representation of a basic cyclic pair. The leaves of form ℓ_* , where $*$ can be a number or a letter, are called the *fixed leaves*. The set of fixed leaves is denoted by $C_{\mathcal{T}}$.

Example 4.6. The set of fixed leaves of the tree representation \mathcal{T} shown in Example 4.5 is $C_{\mathcal{T}} = \{\ell_2\}$.

Definition 4.9. The leaves of the tree \mathcal{T} representing a basic cyclic pair that are not in $C_{\mathcal{T}}$ are called *reticulation leaves*, and the set of reticulation leaves of \mathcal{T} is denoted by $H_{\mathcal{T}}$.

Example 4.7. The set of reticulation leaves of the tree representation \mathcal{T} shown in Example 4.5 is $H_{\mathcal{T}} = \{v_1^{e_1}, v_1^{e_2}, v_2^{e_1}, v_2^{e_2}, v_3^{e_1}, v_3^{e_2}\}$.

Definition 4.10. A switching of the tree \mathcal{T} representing a basic cyclic pair is a leaf restriction R of \mathcal{T} such that $C_{\mathcal{T}} \subseteq R$ and, for every v_i , where $i \in \mathbb{N}$, that is in the set of reticulation vertices of the basic cyclic pair, exactly one of $v_i^{e_1}, v_i^{e_2}$ is in R .

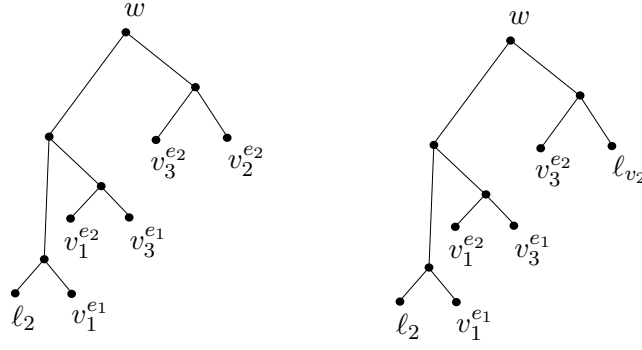
Example 4.8. The leaf restriction $R = \{\ell_2, v_1^{e_1}, v_2^{e_2}, v_3^{e_1}\}$ is a switching of the tree representation \mathcal{T} shown in Example 4.5.

Definition 4.11. A general switching of the tree \mathcal{T} representing a basic cyclic pair $B-(w, v, P_1, P_2)$ is a leaf restriction G of \mathcal{T} such that $C_{\mathcal{T}} \subseteq G$ and for every v_i , where $i \in \mathbb{N}$, that is in the set of reticulation vertices of the basic cyclic pair either exactly one of $v_i^{e_1}, v_i^{e_2}$ is in G or $\{v_i^{e_1}, v_i^{e_2}\} \subseteq G$; that is, G contains at least one of $v_i^{e_1}, v_i^{e_2}$.

Example 4.9. The leaf restriction $G = \{\ell_2, v_1^{e_1}, v_1^{e_2}, v_2^{e_2}, v_3^{e_1}, v_3^{e_2}\}$ is a general switching of the tree representation \mathcal{T} shown in Example 4.5.

Definition 4.12. The tree \mathcal{T} representing a basic cyclic pair displays a tree when there exists a general switching G of \mathcal{T} such that after G is applied to \mathcal{T} every leaf $v_i^{e_j}$, where $j \in \{1, 2\}$ and $i \in \mathbb{N}$, where exactly one of $v_i^{e_1}, v_i^{e_2}$ is in G is relabeled ℓ_{v_i} . We denote applying G to \mathcal{T} by $\mathcal{T}|G$, and we denote the result of the relabeling that takes place afterwards by $(\mathcal{T}|G)^L$.

Example 4.10. The tree $\mathcal{T}|G$ is shown on the left where \mathcal{T} is the tree representation from Example 4.5 and G is from Example 4.9. The tree $(\mathcal{T}|G)^L$ is shown on the right.



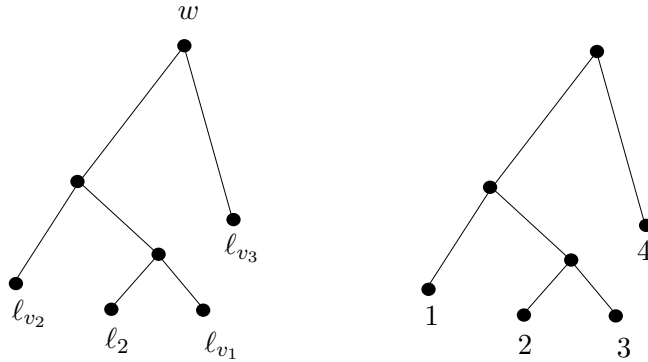
Note 4. Let X be a set of reticulation leaves (X could be a general switching, a switching, or a leaf set) of the tree \mathcal{T} representing a basic cyclic pair then $(X)^L$ means that we relabel every reticulation leaf $v_i^{e_j}$ in X as ℓ_{v_i} .

Example 4.11. Taking G from Example 4.9 we have $(G)^L = \{\ell_2, \ell_{v_1}, \ell_{v_2}, \ell_{v_3}\}$.

Definition 4.13. The tree \mathcal{T} representing a basic cyclic pair $B-(w, v, P_1, P_2)$ displays a tree twice when there exists two switchings R_1, R_2 of \mathcal{T} such that either $v^{e_1} \in R_1$ and $v^{e_2} \in R_2$ or $v^{e_2} \in R_1$ and $v^{e_1} \in R_2$, and $(\mathcal{T}|R_1)^L = (\mathcal{T}|R_2)^L$.

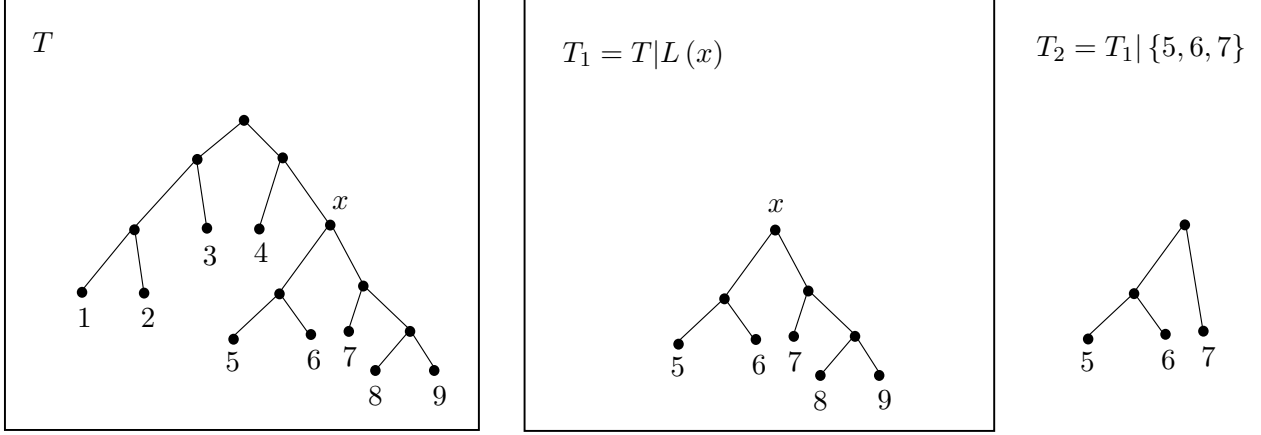
Note 5. We abuse notation and put “=” when we mean “isomorphic to” (in the phylogenetic sense).

Example 4.12. The tree on the left was obtained from Example 4.5 via the switchings $R_1 = \{v_1^{e_1}, v_2^{e_1}, v_3^{e_1}, \ell_2\}$ and $R_2 = \{v_1^{e_1}, v_2^{e_1}, v_3^{e_2}, \ell_2\}$. Hence \mathcal{T} from Example 4.5 representing $B-(w, v_3, P_1, P_2)$ displays the tree on the left twice. The tree on the right was obtained from Example 4.3. It is a subtree of the tree displayed twice by $B-(w, v_3, P_1, P_2)$, and we see that it is simply the tree on the left with its leaves relabeled.



Definition 4.14. Let T be a tree on the leaf set X . A tree T_1 on a leaf set $X_1 \subseteq X$ is a *subtree* of T if there exists a vertex x of T such that $T_1 = T|L(x)$, where $L(x)$ is the set of leaves $\ell \in X$ such that there is a directed path from x to ℓ in T .

Example 4.13. The tree T on the left has leaf set $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. The tree T_1 in the middle is a subtree of T , where $L(x) = \{5, 6, 7, 8, 9\}$. The tree T_2 on the right is not a subtree of T nor a subtree of T_1 .



Lemma 4.1. Let T_1 and T_2 be two trees on the same leaf set X . Let T'_1 be a subtree of T_1 on leaf set X' and let T'_2 be a subtree of T_2 on leaf set X' . If $T'_1 \neq T'_2$ then $T_1 \neq T_2$.

Proof. Suppose $T_1 = T_2$ then it is fairly clear that any subtree of T_1 on a subset of the leaf set X will be isomorphic (in the phylogenetic sense) to the subtree of T_2 on the corresponding leaf set. \square

Note 6. This lemma does not work for multi-labeled trees.

Theorem 4.1. Let \mathcal{T} be the representation of a basic cyclic pair $B-(w, v, P_1, P_2)$. The tree \mathcal{T} displays a tree twice if and only if $B-(w, v, P_1, P_2)$ displays a tree twice.

Proof. Suppose the tree \mathcal{T} displays a tree twice then there exists two switchings R_1, R_2 of \mathcal{T} such that either $v^{e_1} \in R_1$ and $v^{e_2} \in R_2$ or $v^{e_2} \in R_1$ and $v^{e_1} \in R_2$, and $(\mathcal{T}|R_1)^L = (\mathcal{T}|R_2)^L$. Without loss of generality, suppose that $v^{e_1} \in R_1$ and $v^{e_2} \in R_2$. Consider the two set S_1, S_2 where $e_j^{v_i} \in S_1$, for $j \in \{1, 2\}$ and $i \in \mathbb{N}$, if and only if $v_i^{e_j} \in R_1$ and $e_j^{v_i} \in S_2$ if and only if $v_i^{e_j} \in R_2$. Since $v^{e_1} \in R_1$ and $v^{e_2} \in R_2$, we have $e_1^v \in S_1$ and $e_2^v \in S_2$, so that v 's reticulation edges are split between S_1 and S_2 . Now S_1 and S_2 are switchings of $B-(w, v, P_1, P_2)$ because R_1 and R_2 are switchings of \mathcal{T} . We will show that S_1 and S_2 yield the same tree by supposing that they do not and reaching a contradiction. First relabel every leaf n in $B-(w, v, P_1, P_2)$ with ℓ_n , call the result $[B-(w, v, P_1, P_2)]^*$. Suppose that S_1 and S_2 yield different trees, call them T_1 and T_2 respectively. Since every directed path from the root to a reticulation vertex passes through w and that $T_1 \neq T_2$, there exists T'_1 a subtree of T_1 whose leaf set is a subset of the cluster below w in $[B-(w, v, P_1, P_2)]^*$ and there exists T'_2 a subtree of T_2 whose leaf set is the same as the leaf set of T'_1 , and $T'_1 \neq T'_2$. For every vertex x in

T'_1 with the property that there exists a reticulation vertex v_i in $[B-(w, v, P_1, P_2)]^*$ with the same cluster as x , replace x and everything below it by ℓ_{v_i} . Do the same for T'_2 , call the results $(T'_1)^*$ and $(T'_2)^*$ respectively. We must have $(T'_1)^* \neq (T'_2)^*$ and they are on the same leaf set, but $(T'_1)^*$ is a subtree of $(\mathcal{T}|R_1)^L$ and $(T'_2)^*$ is a subtree of $(\mathcal{T}|R_2)^L$; hence, by Lemma 4.1, $(\mathcal{T}|R_1)^L \neq (\mathcal{T}|R_2)^L$ - a contradiction. Therefore S_1 and S_2 yield the same tree. Thus $B-(w, v, P_1, P_2)$ displays a tree twice.

Suppose that $B-(w, v, P_1, P_2)$ displays a tree twice then there exists two switchings S_1, S_2 of $B-(w, v, P_1, P_2)$ such that either $e_1^v \in S_1$ and $e_2^v \in S_2$ or $e_2^v \in S_1$ and $e_1^v \in S_2$, and S_1 and S_2 yield the same tree, call it T^* . Without loss of generality, suppose that $e_1^v \in S_1$ and $e_2^v \in S_2$. Consider the two sets R'_1, R'_2 where $v_i^{e_j} \in R'_1$, where $j \in \{1, 2\}$ and $i \in \mathbb{N}$, if and only if $e_j^{v_i} \in S_1$ and $v_i^{e_j} \in R'_2$ if and only if $e_j^{v_i} \in S_2$. We see that $v^{e_1} \in R'_1$ and $v^{e_2} \in R'_2$. Of course, $R_1 = R'_1 \cup C_{\mathcal{T}}$ and $R_2 = R'_2 \cup C_{\mathcal{T}}$ are switchings of \mathcal{T} . First relabel every fixed leaf ℓ_n in \mathcal{T} with n . Suppose for contradiction that $(\mathcal{T}|R_1)^L \neq (\mathcal{T}|R_2)^L$. Now $(\mathcal{T}|R_1)^L$ and $(\mathcal{T}|R_2)^L$ are on the same leaf set, because take any ℓ_{v_i} in the leaf set of $(\mathcal{T}|R_1)^L$ then either $v_i^{e_1}$ or $v_i^{e_2}$ is in R_1 . Since R_2 is a switching, exactly one of $v_i^{e_1}, v_i^{e_2}$ is in R_2 , so that ℓ_{v_i} is in the leaf set of $(\mathcal{T}|R_2)^L$. Similarly for the other direction. Since $B-(w, v, P_1, P_2)$ is a basic cyclic pair, for every reticulation vertex v_i in $B-(w, v, P_1, P_2)$ there is a tree T_{v_i} whose root is the child of v_i . In $(\mathcal{T}|R_1)^L$ replace every ℓ_{v_i} with the tree T_{v_i} , call the result T_1^* . Do the same in $(\mathcal{T}|R_2)^L$, call the result T_2^* . We must have $T_1^* \neq T_2^*$, but we also have T_1^*, T_2^* being two subtrees of T^* on the same leaf set, so, by Lemma 4.1, $T_1^* = T_2^*$ - a contradiction. Therefore $(\mathcal{T}|R_1)^L = (\mathcal{T}|R_2)^L$, so that \mathcal{T} displays a tree twice. \square

4.4 Reducing a Basic Cyclic Pair

One of the most important results for deciding quickly whether or not an arbitrary phylogenetic network displays a tree twice is the following proposition. If a basic cyclic pair displays a tree twice then any reduction of the basic cyclic pair displays a tree twice. This proposition implies that if we are able to reduce a basic cyclic pair to a basic cyclic pair that does not display a tree twice then the original basic cyclic pair does not display a tree twice. We shall see that there are certain archetypal basic cyclic pairs that never display a tree twice, and it is these basic cyclic pairs, which we call desirable configurations, that makes the task of deciding quickly whether or not an arbitrary phylogenetic network displays a tree twice possible. To reduce a basic cyclic pair $B-(w, v, P_1, P_2)$ is to reduce the representation of the basic cyclic pair, and to reduce the representation of the basic cyclic pair is to apply a leaf restriction that keeps the leaves v^{e_1}, v^{e_2} (we may discard any fixed leaves), but to keep $v_i^{e_1}$ we must also keep $v_i^{e_2}$. The proof of the reduction proposition depends on

two lemmas. The first lemma says that it does not matter what order two leaf restrictions are done in, and it does not matter how a particular leaf restriction is performed. The second lemma says that if we have a leaf restriction that is a switching and another leaf restriction that is a reduction then when we relabel does not affect the result, so that if we relabel after the switching is applied and then apply a relabeled reduction then that is the same as relabeling after applying the switching and the reduction.

Definition 4.15. A *reduction* of a basic cyclic pair $B-(w, v, P_1, P_2)$ is a leaf restriction Γ of the tree representation \mathcal{T} of $B-(w, v, P_1, P_2)$ such that $\{v^{e_1}, v^{e_2}\} \subseteq \Gamma$ and for all v_i in $B-(w, v, P_1, P_2)$, $v_i^{e_j} \in \Gamma$ if and only if $v_i^{e_{\{1,2\} \setminus j}} \in \Gamma$.

Example 4.14. The leaf restriction $\Gamma = \{v_3^{e_1}, v_3^{e_2}, v_1^{e_1}, v_1^{e_2}, \ell_2\}$ is a reduction of the basic cyclic pair $B-(w, v_3, P_1, P_2)$ from Example 4.3, but it is not a reduction of the basic cyclic pair $B-(w, v_2, P_1, P_2)$.

Lemma 4.2. Let \mathcal{T} be the representation of a basic cyclic pair $B-(w, v, P_1, P_2)$. Let R be a switching of \mathcal{T} , and let Γ be a reduction of $B-(w, v, P_1, P_2)$. Then $(\mathcal{T}|R)|\Gamma = (\mathcal{T}|R \cap \Gamma)$ and $(\mathcal{T}|\Gamma)|R = (\mathcal{T}|R \cap \Gamma)$.

Proof. It does not matter what order two restrictions are done in, so $(\mathcal{T}|\Gamma)|R = (\mathcal{T}|R)|\Gamma$. Now $(\mathcal{T}|R \cap \Gamma)$ takes \mathcal{T} and restricts \mathcal{T} to the leaves that are in both R and Γ ; note that $R \cap \Gamma \subseteq X_{\mathcal{T}}$, where $X_{\mathcal{T}}$ is the leaf set of the tree \mathcal{T} . Also $(\mathcal{T}|R)|\Gamma$ takes \mathcal{T} and first restricts it to the leaves in R then restricts $\mathcal{T}|R$ to the leaves in Γ . This gives the same result as $(\mathcal{T}|R \cap \Gamma)$. Note that the leaf set of $\mathcal{T}|R$ is $X_{\mathcal{T}} \cap R$, so that the leaf set of $(\mathcal{T}|R)|\Gamma$ is $X_{\mathcal{T}} \cap R \cap \Gamma$, which is the same as the leaf set of $(\mathcal{T}|R \cap \Gamma)$. Therefore $(\mathcal{T}|R)|\Gamma = (\mathcal{T}|R \cap \Gamma)$ and $(\mathcal{T}|\Gamma)|R = (\mathcal{T}|R \cap \Gamma)$. \square

Lemma 4.3. Let \mathcal{T} be the representation of a basic cyclic pair $B-(w, v, P_1, P_2)$. Let R be a switching of \mathcal{T} , and let Γ be a reduction of $B-(w, v, P_1, P_2)$. Then $((\mathcal{T}|R)|\Gamma)^L = (\mathcal{T}|R)^L | (\Gamma)^L$.

Proof. The leaf set of $((\mathcal{T}|R)|\Gamma)^L$ is $(X_{\mathcal{T}} \cap R \cap \Gamma)^L$. We will show that $(X_{\mathcal{T}} \cap R \cap \Gamma)^L = (X_{\mathcal{T}} \cap R)^L \cap (\Gamma)^L$. Take any element in $(X_{\mathcal{T}} \cap R \cap \Gamma)^L$ then it is of form ℓ_* , so either $*^{e_1} \in X_{\mathcal{T}} \cap R \cap \Gamma$, $*^{e_2} \in X_{\mathcal{T}} \cap R \cap \Gamma$, or $\ell_* \in X_{\mathcal{T}} \cap R \cap \Gamma$. If $\ell_* \in X_{\mathcal{T}} \cap R \cap \Gamma$ then $\ell_* \in (X_{\mathcal{T}} \cap R)^L$ and $\ell_* \in (\Gamma)^L$, so $\ell_* \in (X_{\mathcal{T}} \cap R)^L \cap (\Gamma)^L$. Without loss of generality, suppose that $*^{e_1} \in X_{\mathcal{T}} \cap R \cap \Gamma$ then $*^{e_1} \in X_{\mathcal{T}} \cap R$ and $*^{e_1} \in \Gamma$, so $\ell_* \in (X_{\mathcal{T}} \cap R)^L$ and $\ell_* \in (\Gamma)^L$; hence $\ell_* \in (X_{\mathcal{T}} \cap R)^L \cap (\Gamma)^L$. Take any element in $(X_{\mathcal{T}} \cap R)^L \cap (\Gamma)^L$ then it is of form ℓ_* , so $\ell_* \in (X_{\mathcal{T}} \cap R)^L$ and $\ell_* \in (\Gamma)^L$. If $\ell_* \in X_{\mathcal{T}}$ then $\ell_* \in X_{\mathcal{T}} \cap R$ and $\ell_* \in \Gamma$; hence $\ell_* \in (X_{\mathcal{T}} \cap R \cap \Gamma)$, so $\ell_* \in (X_{\mathcal{T}} \cap R \cap \Gamma)^L$. If $\ell_* \notin X_{\mathcal{T}}$ then, since Γ is a reduction, $\{*^{e_1}, *^{e_2}\} \subseteq \Gamma$, as $\ell_* \in (\Gamma)^L$. Since R is a switching, $\ell_* \in (X_{\mathcal{T}} \cap R)^L$, and $\ell_* \notin X_{\mathcal{T}}$, we have either $*^{e_1} \in X_{\mathcal{T}} \cap R$ or $*^{e_2} \in X_{\mathcal{T}} \cap R$. Without loss of generality, suppose that $*^{e_1} \in X_{\mathcal{T}} \cap R$

then $*^{e_1} \in X_{\mathcal{T}} \cap R \cap \Gamma$, so $\ell_* \in (X_{\mathcal{T}} \cap R \cap \Gamma)^L$. Therefore $(X_{\mathcal{T}} \cap R \cap \Gamma)^L = (X_{\mathcal{T}} \cap R)^L \cap (\Gamma)^L$. Now the leaf set of $(\mathcal{T}|R)^L$ is $(X_{\mathcal{T}} \cap R)^L$ and the leaf set of $(\mathcal{T}|R)^L | (\Gamma)^L$ is $(X_{\mathcal{T}} \cap R)^L \cap (\Gamma)^L$, so $((\mathcal{T}|R) | \Gamma)^L$ has the same leaf set as $(\mathcal{T}|R)^L | (\Gamma)^L$. Take $(\mathcal{T}|R)$ and relabel every $v_i^{e_j}$ as ℓ_{v_i} to get $(\mathcal{T}|R)^L$ then restrict $(\mathcal{T}|R)^L$ to $(\Gamma)^L$. This is the same as taking $(\mathcal{T}|R)$ restricting to Γ then relabeling every $v_i^{e_j}$ as ℓ_{v_i} . Thus $((\mathcal{T}|R) | \Gamma)^L = (\mathcal{T}|R)^L | (\Gamma)^L$. \square

Proposition 4.1. *Let $B-(w, v, P_1, P_2)$ be a basic cyclic pair. If $B-(w, v, P_1, P_2)$ displays a tree twice then any reduction of $B-(w, v, P_1, P_2)$ displays a tree twice.*

Proof. Suppose $B-(w, v, P_1, P_2)$ displays a tree twice then the tree \mathcal{T} representing $B-(w, v, P_1, P_2)$ displays a tree twice, so there exists two switchings R_1, R_2 of \mathcal{T} such that either $v^{e_1} \in R_1$ and $v^{e_2} \in R_2$ or $v^{e_2} \in R_1$ and $v^{e_1} \in R_2$, and $(\mathcal{T}|R_1)^L = (\mathcal{T}|R_2)^L$. Let Γ be any reduction of $B-(w, v, P_1, P_2)$, so Γ is a leaf restriction of \mathcal{T} where for every $\{v_i^{e_1}, v_i^{e_2}\} \subseteq H_{\mathcal{T}}$ either $\{v_i^{e_1}, v_i^{e_2}\} \subseteq \Gamma$ or $\{v_i^{e_1}, v_i^{e_2}\} \cap \Gamma = \emptyset$, and $\{v^{e_1}, v^{e_2}\} \subseteq \Gamma$. We want to show that $\mathcal{T}|\Gamma$ displays a tree twice. Now the leaf set of $\mathcal{T}|\Gamma$ is

$$X_{\mathcal{T}} \cap \Gamma = (H_{\mathcal{T}} \cup C_{\mathcal{T}}) \cap \Gamma = (H_{\mathcal{T}} \cap \Gamma) \cup (C_{\mathcal{T}} \cap \Gamma).$$

Let R'_1 be $(R_1 \cap \Gamma)$, and let R'_2 be $(R_2 \cap \Gamma)$. We want to show that R'_1 and R'_2 are switchings of $\mathcal{T}|\Gamma$. Since $C_{\mathcal{T}} \subseteq R_1$, we see that $R_1 = (R_1 \cap H_{\mathcal{T}}) \cup C_{\mathcal{T}}$, so

$$R'_1 = R_1 \cap \Gamma = [(R_1 \cap H_{\mathcal{T}}) \cup C_{\mathcal{T}}] \cap \Gamma.$$

Hence $C_{\mathcal{T}} \cap \Gamma \subseteq R'_1$. Since R_1 is a switching of \mathcal{T} , for every $\{v_i^{e_1}, v_i^{e_2}\} \subseteq H_{\mathcal{T}}$, where $i \in \mathbb{N}$, exactly one of $v_i^{e_1}, v_i^{e_2}$ is in R_1 , so for every $\{v_i^{e_1}, v_i^{e_2}\} \subseteq (H_{\mathcal{T}} \cap \Gamma)$ exactly one of $v_i^{e_1}, v_i^{e_2}$ is in R'_1 . Hence for every $\{v_i^{e_1}, v_i^{e_2}\} \subseteq (H_{\mathcal{T}} \cap \Gamma)$ exactly one of $v_i^{e_1}, v_i^{e_2}$ is in R'_1 . Thus R'_1 is a switching of $\mathcal{T}|\Gamma$. Likewise R'_2 is a switching of $\mathcal{T}|\Gamma$. Without loss of generality, suppose that $v^{e_1} \in R_1$ and $v^{e_2} \in R_2$, since $\{v^{e_1}, v^{e_2}\} \subseteq \Gamma$, we have $\{v^{e_1}, v^{e_2}\} \subseteq (H_{\mathcal{T}} \cap \Gamma)$, so that $v^{e_1} \in R'_1$ and $v^{e_2} \in R'_2$. Consider $((\mathcal{T}|\Gamma) | R'_1)^L$ and $((\mathcal{T}|\Gamma) | R'_2)^L$. By Lemma 4.2, the order in which leaf restrictions are done in does not matter, so $((\mathcal{T}|\Gamma) | R'_1)^L = ((\mathcal{T}|R'_1) | \Gamma)^L$. We also have, by Lemma 4.3, $((\mathcal{T}|R'_1) | \Gamma)^L = (\mathcal{T}|R'_1)^L | (\Gamma)^L$. Now $(\mathcal{T}|R'_1) | \Gamma = (\mathcal{T}|R'_1)$, so $((\mathcal{T}|R'_1) | (\Gamma))^L = (\mathcal{T}|R'_1)^L$. Likewise we can show that $((\mathcal{T}|R'_2) | (\Gamma))^L = (\mathcal{T}|R'_2)^L$. Since $(\mathcal{T}|R_1)^L = (\mathcal{T}|R_2)^L$, we have $(\mathcal{T}|R_1)^L | (\Gamma)^L = (\mathcal{T}|R_2)^L | (\Gamma)^L$. By Lemma 4.3, $(\mathcal{T}|R_1)^L | (\Gamma)^L = ((\mathcal{T}|R_1) | \Gamma)^L$, and, by Lemma 4.2, $((\mathcal{T}|R_1) | \Gamma)^L = (\mathcal{T}|R_1 \cap \Gamma)^L$. Of course, $(\mathcal{T}|R_1 \cap \Gamma)^L = (\mathcal{T}|R'_1)^L$. Likewise we can show that $(\mathcal{T}|R_2)^L | (\Gamma)^L = (\mathcal{T}|R'_2)^L$. Therefore $(\mathcal{T}|R'_1)^L = (\mathcal{T}|R'_2)^L$. Thus we get $((\mathcal{T}|\Gamma) | R'_1)^L = ((\mathcal{T}|\Gamma) | R'_2)^L$, and that means that $\mathcal{T}|\Gamma$ displays a tree twice via R'_1 and R'_2 . Therefore any reduction of $B-(w, v, P_1, P_2)$ displays a tree twice. \square

Definition 4.16. Let \mathcal{T} be the representation of a basic cyclic pair $B-(w, v, P_1, P_2)$. We say that \mathcal{T} is a *desirable configuration* if \mathcal{T} does not display a tree twice.

Corollary 4.1. *Let $B-(w, v, P_1, P_2)$ be a basic cyclic pair. The basic cyclic pair $B-(w, v, P_1, P_2)$ displays a tree twice if and only if $B-(w, v, P_1, P_2)$ can not be reduced to a desirable configuration.*

Proof. Suppose $B-(w, v, P_1, P_2)$ displays a tree twice then, by Proposition 4.1, any reduction of $B-(w, v, P_1, P_2)$ displays a tree twice. Therefore $B-(w, v, P_1, P_2)$ can not be reduced to a desirable configuration.

Suppose that $B-(w, v, P_1, P_2)$ can not be reduced to a desirable configuration then every reduction of $B-(w, v, P_1, P_2)$ results in a basic cyclic pair that displays a tree twice. Therefore the trivial reduction of $B-(w, v, P_1, P_2)$ displays a tree twice. Thus $B-(w, v, P_1, P_2)$ displays a tree twice. \square

4.5 Avoidable Reticulation Vertices

The characterisation of when a phylogenetic network displays a tree twice only characterises a subclass of phylogenetic networks. This subclass of phylogenetic networks are all free of a certain type of reticulation vertex, which is called an avoidable reticulation vertex. If a phylogenetic network has an avoidable reticulation vertex then the network displays a tree twice, so every network outside of the subclass of networks without avoidable reticulation vertices displays a tree twice. We will show that the property of not having any avoidable reticulation vertices is preserved under general switchings, so if a network does not have any avoidable reticulation vertices then any network displayed by the network does not have any avoidable reticulation vertices.

Lemma 4.4. *Let N be a network on X . Let G be a general switching of N , and let N' be the network yielded by G . If N does not have any avoidable reticulation vertices then N' does not have any avoidable reticulation vertices.*

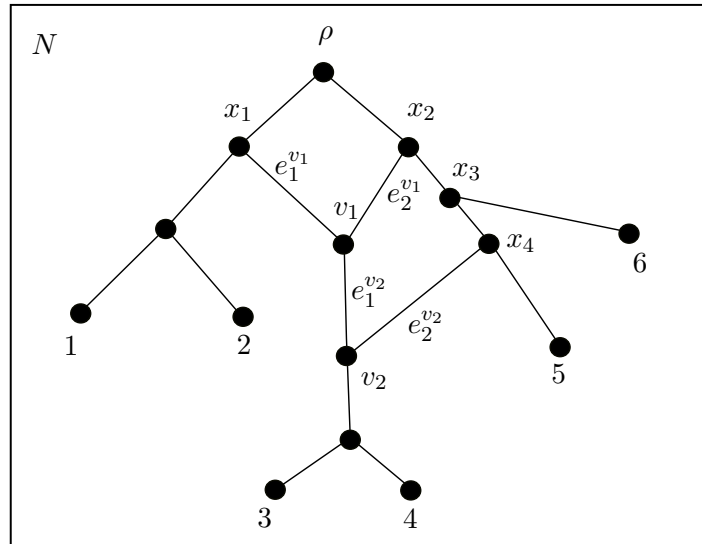
Proof. Suppose that N does not have any avoidable reticulation vertices. Let v be any reticulation vertex in N' then v is a reticulation vertex in N ; thus v is unavoidable in N , so there exists a leaf $\ell \in X$ such that every directed path from the root of N to ℓ passes through v . Since N' is a network on X , the leaf ℓ is in N' . Let P' be any directed path from the root of N' to ℓ in N' . We see that there is a directed path P from the root of N to ℓ in N such that $V(P') \subseteq V(P)$. Since every directed path from the root of N to ℓ in N passes through v , we see that P' passes through v . Therefore v is unavoidable in N' . \square

4.6 Characterising when a Phylogenetic Network Displays a Tree Twice in Terms of Cyclic Pairs in the Network

We have already seen that a network with an avoidable reticulation vertex displays a tree twice, so we now characterise when networks without any avoidable reticulation vertices display a tree twice. The advantage of not having any avoidable reticulation vertices in a network is that we are not going to inadvertently lose a reticulation vertex when a general switching is applied to the network. In order to get the characterisation that leads to an efficient algorithm we need a few stepping stones. The first stepping stone that we need is another type of cyclic pair, and it is found by seeing what it means for a phylogenetic network, without any avoidable reticulation vertices, to display a tree twice. This other type of cyclic pair has no reticulation vertices on its main paths and if one can get from the source of the cyclic pair to a reticulation vertex via a tree-path then there are two tree-paths that differ at at least one edge from the source to that reticulation vertex.

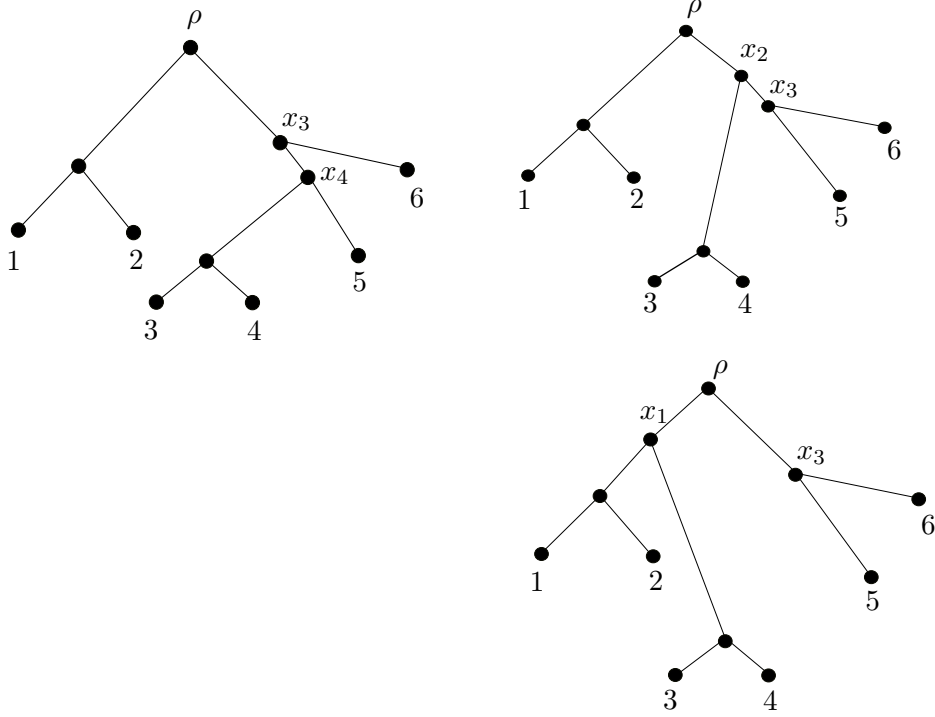
Definition 4.17. Let N be a network, and let w be a tree-vertex in N . Let v be a reticulation vertex in N such that there exists two directed paths P_1, P_2 such that $E(P_1) \cap E(P_2) = \emptyset$ and $V(P_1) \cap V(P_2) = \{w, v\}$. The ordered tuple (w, v, P_1, P_2) and the network N is called a *cyclic pair*. We denote a cyclic pair by $N-(w, v, P_1, P_2)$. We also call w the *source with respect to* $N-(w, v, P_1, P_2)$, and we call v the *sink with respect to* $N-(w, v, P_1, P_2)$.

Example 4.15. There are three cyclic pairs in the network N shown, namely: $N-(\rho, v_1, P_1, P_2)$, where $P_1 = \{(\rho, x_1), e_1^{v_1}\}$ and $P_2 = \{(\rho, x_2), e_2^{v_1}\}$; $N-(\rho, v_2, P'_1, P'_2)$, where $P'_1 = \{(\rho, x_1), e_1^{v_1}, e_1^{v_2}\}$ and $P'_2 = \{(\rho, x_2), (x_2, x_3), (x_3, x_4), e_2^{v_2}\}$; and $N-(x_2, v_2, P''_1, P''_2)$, where $P''_1 = \{e_2^{v_1}, e_1^{v_2}\}$ and $P''_2 = \{(x_2, x_3), (x_3, x_4), e_2^{v_2}\}$.



Definition 4.18. A cyclic pair $N-(w, v, P_1, P_2)$ displays a tree twice when there exists two switchings S_1, S_2 of N that yield the same tree, and one of v 's reticulation edges is in S_1 whilst the other is in S_2 .

Example 4.16. The cyclic pair $N-(\rho, v_1, P_1, P_2)$ from Example 4.15 displays the tree shown on the left twice via $S_1 = \{e_1^{v_1}, e_1^{v_2}\}$ and $S_2 = \{e_2^{v_1}, e_1^{v_2}\}$. The other cyclic pairs from Example 4.15 do not display the tree on the left twice. The other trees displayed by N are shown on the right.



Definition 4.19. A cyclic pair $N-(w, v, P_1, P_2)$ is a *tree-path cyclic pair* when both P_1 and P_2 are tree-paths.

Example 4.17. The only tree-path cyclic pair from the network N from Example 4.15 is $N-(\rho, v_1, P_1, P_2)$.

Definition 4.20. Let $N-(w, v, P_1, P_2)$ be a cyclic pair. A reticulation vertex v_1 is a *tree-path reticulation with respect to w* if there is a tree-path from w to v_1 .

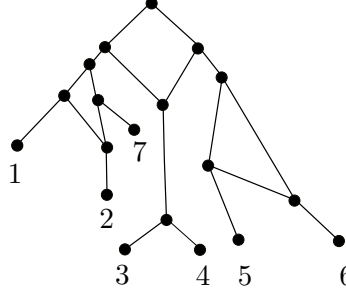
Example 4.18. Both v_1 and v_2 are tree-path reticulations with respect to ρ for the cyclic pairs $N-(\rho, v_1, P_1, P_2)$ and $N-(\rho, v_2, P'_1, P'_2)$ from Example 4.15. Likewise v_1 and v_2 are tree-path reticulations with respect to x_2 for $N-(x_2, v_2, P''_1, P''_2)$ from Example 4.15.

Definition 4.21. Let $N-(w, v, P_1, P_2)$ be a cyclic pair. A tree-path reticulation v_1 with respect to w is a *two-way tree-path reticulation with respect to w* if there are two tree-paths from w to v_1 that differ at at least one edge.

Example 4.19. Only v_1 is a two-way tree-path reticulation with respect to ρ for the cyclic pairs $N-(\rho, v_1, P_1, P_2)$ and $N-(\rho, v_2, P'_1, P'_2)$ from Example 4.15. Neither v_1 nor v_2 are two-way tree-path reticulations with respect to x_2 for the cyclic pair $N-(x_2, v_2, P''_1, P''_2)$ from Example 4.15.

Definition 4.22. A tree-path cyclic pair $N-(w, v, P_1, P_2)$ is a *maximal tree-path cyclic pair* if every tree-path reticulation with respect to w is a two-way tree-path reticulation with respect to w .

Example 4.20. Every cyclic pair in the network shown is a maximal tree-path cyclic pair.



Proposition 4.2. Let N be a network without any avoidable reticulation vertices. Let X be the leaf set of N . If N displays a tree twice then there exists a maximal tree-path cyclic pair, displayed by N , that displays a tree twice.

Proof. Suppose N displays a tree twice then there exists two switchings S_1, S_2 of N such that $S_1 \neq S_2$ and S_1, S_2 yield the same tree when they are applied to N . Since $S_1 \neq S_2$, there exists a non-empty set D of reticulation vertices of N such that $v \in D$ if and only if either $e_1^v \in S_1$ and $e_2^v \in S_2$ or $e_2^v \in S_1$ and $e_1^v \in S_2$. Let the set of reticulation vertices of N be called K . Then $K \setminus D$ is the set of reticulation vertices of N such that $v_1 \in K \setminus D$ if and only if either $e_1^{v_1} \in S_1$ and $e_1^{v_1} \in S_2$ or $e_2^{v_1} \in S_1$ and $e_2^{v_1} \in S_2$. Let $E_{K \setminus D}$ be the set of reticulation edges of N such that $e_i^{v_j} \in E_{K \setminus D}$, where $i \in \{1, 2\}$ and $j \in \mathbb{N}$, if and only if $v_j \in K \setminus D$. We see that $S_1 \cap E_{K \setminus D} = S_2 \cap E_{K \setminus D}$. We can order the set D by descendency, so there exists $v \in D$ such that v has no ancestors in D . Let D_1 be the set of all such reticulation vertices v . Let E_D be the set of reticulation edges of N such that $e_i^{v_j} \in E_D$ if and only if $v_j \in D$. Consider the general switching G defined as $(S_1 \cap E_{K \setminus D}) \cup E_D$. Since N has no avoidable reticulation vertices, every reticulation vertex in D remains after G is applied to N . Let N_G be the network that is the result of applying G to N . In particular, the reticulation vertices in D_1 are in N_G . Let v be any reticulation vertex in D_1 . Since the only reticulation vertices in N_G are those in D and v has no ancestors in D , the two paths starting at v and traveling towards the root of N_G are tree-paths. Let w be the first tree-vertex on those two tree-paths starting at v . Therefore $N_G-(w, v, P_1, P_2)$ is a tree-path cyclic pair. Let K^G be the set of reticulation vertices in

N_G . Let E_{KG} be the set of reticulation edges in N_G . We see that $N_G-(w, v, P_1, P_2)$ displays a tree T twice via $S_1 \cap E_{KG}$ and $S_2 \cap E_{KG}$.

Let J be the set of two-way tree-path reticulation vertices with respect to w in N_G . Let E_J be the set of reticulation edges of N_G such that $e_i^{v_j} \in E_J$ if and only if $v_j \in J$. Let M be the set of tree-path reticulations with respect to w that are not in J . For each $v_j \in M$ exactly one of $e_1^{v_j}, e_2^{v_j}$ is on the tree-path from w to v_j . Let E_{MT} be the set of reticulation edges in N_G such that $e_i^{v_j} \in E_{MT}$ if and only if $v_j \in M$ and $e_i^{v_j}$ is not on a tree-path from w to v_j . Let E_M be the set of reticulation edges in N_G such that $e_i^{v_j} \in E_M$ if and only if $v_j \in M$. Consider the general switching $G_2 = (S_2 \cap (E_{KG} \setminus E_M)) \cup E_J \cup E_{MT}$. We see that applying G_2 to N_G deletes all the reticulation edges that are on a tree-path from w to a reticulation vertex that is not in J , and the only reticulation vertices left are those in J . Let N_{G_2} be the network displayed by N_G via G_2 . We see that there is a maximal tree-path cyclic pair in N_{G_2} whose source is w and whose sink is an element of J . Let $N_{G_2}-(w, v, P'_1, P'_2)$ be such a maximal tree-path cyclic pair in N_{G_2} . Let E_{KG_2} be the set of reticulation edges in N_{G_2} . We claim that $N_{G_2}-(w, v, P'_1, P'_2)$ displays a tree twice via $S_1 \cap E_{KG_2}$ and $S_2 \cap E_{KG_2}$. Let T_1 be the tree displayed by N_{G_2} via $S_1 \cap E_{KG_2}$, and let T_2 be the tree displayed by N_{G_2} via $S_2 \cap E_{KG_2}$. Suppose for contradiction that $T_1 \neq T_2$ then, by Lemma 1.2, there exists leaves $\ell_1, \ell_2, \ell_3 \in X$ such that T_1 contains the triple $\ell_1\ell_2|\ell_3$ whilst T_2 does not, T_1 contains the triple $\ell_1\ell_3|\ell_2$ whilst T_2 does not, or T_1 contains the triple $\ell_2\ell_3|\ell_1$ whilst T_2 does not.

Without loss of generality, suppose that T_1 contains the triple $\ell_1\ell_2|\ell_3$ whilst T_2 does not. In N_{G_2} there must be a reticulation vertex $v_1 \in J$ in N_{G_2} such that there is a tree-path from v_1 to exactly one of ℓ_1, ℓ_2, ℓ_3 . Without loss of generality, suppose there is a tree-path from v_1 to ℓ_1 . Now in N_G v_1 exists, and, since N has no avoidable reticulation vertices, we see that v_1 is not an avoidable reticulation vertex by Lemma 4.4. Hence there exists a leaf ℓ'_1 such that every directed path from the root of N_G to ℓ'_1 passes through v_1 . Coming back to N_{G_2} , we see that T_1 contains the triple $\ell'_1\ell_2|\ell_3$. If in $N_{G_2}-(w, v, P'_1, P'_2)$ there are two tree-paths from w to ℓ_2, ℓ_3 such that the tree-path from w to ℓ_2 leaves P'_1 , say, at a tree-vertex x_1 whilst the tree-path from w to ℓ_3 leaves P'_1 or P'_2 at a tree-vertex x_2 , where $x_1 \neq x_2$, then N_G does not display the tree T twice. For the tree T'_1 displayed by N_G via $S_1 \cap E_{KG}$ contains either the triple $\ell'_1\ell_2|\ell_3$ or the triple $\ell'_1\ell_3|\ell_2$ or the triple $\ell_2\ell_3|\ell'_1$. Without loss of generality, suppose that T'_1 contains the triple $\ell'_1\ell_2|\ell_3$ then the tree T'_2 displayed by N_G via $S_2 \cap E_{KG}$ contains either the triple $\ell'_1\ell_3|\ell_2$ or the triple $\ell_2\ell_3|\ell'_1$, so that $T'_1 \neq T'_2$ - a contradiction. Hence in N_{G_2} there must be another reticulation vertex v_2 such that there is a tree-path from v_2 to exactly one of ℓ_2 or ℓ_3 . Without loss of generality, suppose that there is a tree-path from v_2 to ℓ_2 . Then, as before with v_1 so too with v_2 , there exists a leaf ℓ'_2 such that

every directed path from the root of N_G to ℓ'_2 passes through v_2 . We see that T_1 contains the triple $\ell'_1\ell'_2|\ell_3$, and suppose that T_2 contains the triple $\ell'_1\ell_3|\ell'_2$. For now suppose that there is a tree-path from w to ℓ_3 in N_{G_2} then there is a tree-path from w to ℓ_3 in N_G . Suppose that T contains the triple $\ell'_1\ell'_2|\ell_3$ then there is a path connecting ℓ'_1 and ℓ'_2 that does not intersect the directed path from w to ℓ_3 in N_G . Hence there is a reticulation vertex v_1^* on the said path connecting ℓ'_1 and ℓ'_2 such that either ℓ'_1 is a descendant of v_1^* or ℓ'_2 is a descendant of v_1^* . Without loss of generality, suppose that ℓ'_1 is a descendant of v_1^* . Since $v_1 \in J$, we must have v_1^* being a descendant of v_1 , as otherwise $v_1 \notin J$ - a contradiction. Now we see that there is a directed path from the root of N_G to ℓ'_1 that avoids v_1 - a contradiction. Therefore, in this case, we reach a contradiction.

Suppose that T contains the triple $\ell'_1\ell_3|\ell'_2$ then there is a path connecting ℓ'_1 and ℓ_3 that does not intersect the directed path from w to ℓ'_2 in N_G . Hence there is a reticulation vertex v_3^* on the said path connecting ℓ'_1 and ℓ_3 such that either ℓ'_1 is a descendant of v_3^* or ℓ_3 is a descendant of v_3^* . The case where ℓ'_1 is a descendant of v_3^* leads to a contradiction as in the last case, so suppose that ℓ_3 is a descendant of v_3^* . Then we see that there is no tree-path from w to ℓ_3 in N_G - a contradiction. Suppose that T contains the triple $\ell'_2\ell_3|\ell'_1$ then we reach two contradictions. Of course, if we suppose that there is no tree-path from w to ℓ_3 in N_{G_2} then we still reach a contradiction. Therefore $T_1 = T_2$, so that $N_{G_2}-(w, v, P'_1, P'_2)$ displays a tree twice via $S_1 \cap E_{K^{G_2}}$ and $S_2 \cap E_{K^{G_2}}$. \square

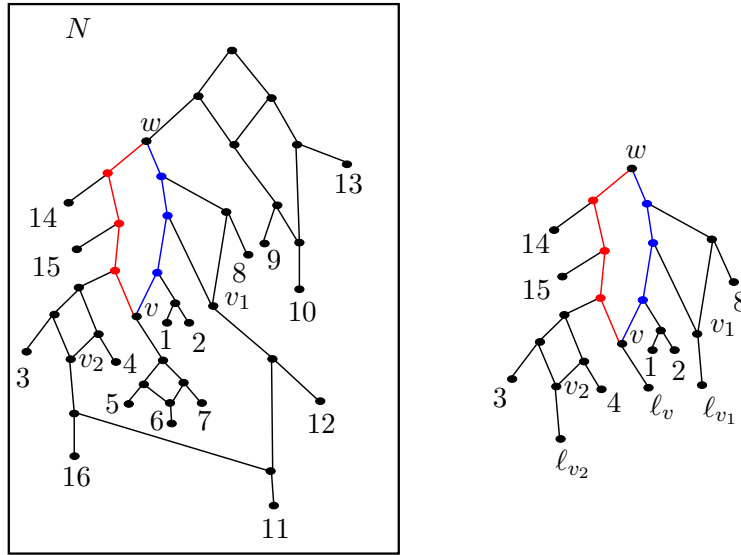
4.6.1 Associated basic cyclic pairs

The next task is to build a bridge between basic cyclic pairs and these maximal tree-path cyclic pairs. A new concept, that of an associated basic cyclic pair of a maximal tree-path cyclic pair, is the bridge between basic cyclic pairs and maximal tree-path cyclic pairs. The associated basic cyclic pair of a maximal tree-path cyclic pair is obtained from the maximal tree-path cyclic pair by the following procedure: Make every reticulation vertex in the maximal tree-path cyclic pair adjacent to a leaf. Delete the directed edge going into the source with respect to the maximal tree-path cyclic pair. Then delete any components that do not contain the source with respect to the maximal tree-path cyclic pair. The result is a basic cyclic pair, and we call this the associated basic cyclic pair of the maximal tree-path cyclic pair. We will then prove that a maximal tree-path cyclic pair displays a tree twice if and only if its associated basic cyclic pair displays a tree twice with the help of a lemma. The lemma says that if a network displayed by a network displays a tree twice then the original network displays a tree twice.

Definition 4.23. Let $N-(w, v, P_1, P_2)$ be a maximal tree-path cyclic pair. The *associated basic cyclic pair* $B-(w, v, P_1, P_2)$ of $N-(w, v, P_1, P_2)$ is obtained from $N-(w, v, P_1, P_2)$ as follows:

1. For each reticulation vertex v_i that is a two-way tree-path reticulation with respect to w in N , where $i \in A$ and A is a finite subset of the natural numbers, delete the edge going out of v_i .
2. Create a new leaf (call it ℓ_{v_i}), and make v_i adjacent to ℓ_{v_i} .
3. Delete the edge going into w .
4. Delete any components that do not contain w .

Example 4.21. The network N shown has the maximal tree-path cyclic pair $N-(w, v, P_1, P_2)$, where P_1 is shown in red and P_2 is shown in blue. Shown on the right is the associated basic cyclic pair of $N-(w, v, P_1, P_2)$.



Lemma 4.5. *Let N be a network without any avoidable reticulation vertices. Let G be a general switching of N , and let N' be the network yielded by G . If N' displays a tree twice then N displays a tree twice. Moreover, let S'_1 and S'_2 be the two switchings of N' that yield the same tree then there exists two switchings S_1 and S_2 of N such that $S'_1 \subseteq S_1$, $S'_2 \subseteq S_2$, and S_1, S_2 yield the same tree.*

Proof. Suppose that N' displays a tree twice then there exists two switchings S'_1, S'_2 such that $S'_1 \neq S'_2$ and S'_1, S'_2 yield the same tree. Let K' be the set of reticulation vertices in N' . Then, since S'_1 is a switching of N' , for every $v' \in K'$ exactly one of $e_1^{v'}, e_2^{v'}$ is in S'_1 . Similarly for S'_2 . Consider the two switchings S_1, S_2 of N where $S'_1 \subseteq S_1 \subseteq G$ and $S'_2 \subseteq S_2 \subseteq G$. It is fairly clear that we can not have $e_1^v \in S'_1$ and $e_1^v \notin G$ nor can we have $e_2^v \in S'_1$ and $e_2^v \notin G$ for any reticulation vertex v in N . Therefore S_1 being a switching and $S'_1 \subseteq S_1 \subseteq G$ are compatible. Likewise with S_2 . Applying S_1 to N is the same as applying G to N , but not deleting or suppressing any vertices, and then

applying S'_1 . It makes no difference whether or not we delete and suppress vertices as we go or leave it till the end. Therefore the result of applying S_1 to N is the tree that is yielded by S'_1 when S'_1 is applied to N' . Similarly for S_2 . Thus S_1 and S_2 yield the same tree, as S'_1, S'_2 yield the same tree. Since $S'_1 \neq S'_2$, $S'_1 \subseteq S_1$, and $S'_2 \subseteq S_2$, we have $S_1 \neq S_2$. Therefore N displays a tree twice. \square

Proposition 4.3. *Let N be a network without any avoidable reticulation vertices, and let $N-(w, v, P_1, P_2)$ be a maximal tree-path cyclic pair. The maximal tree-path cyclic pair $N-(w, v, P_1, P_2)$ displays a tree twice if and only if its associated basic cyclic pair $B-(w, v, P_1, P_2)$ displays a tree twice.*

Proof. Suppose that $N-(w, v, P_1, P_2)$ displays a tree twice then there exists two switchings S_1, S_2 such that, without loss of generality, $e_1^v \in S_1$ and $e_2^v \in S_2$, and S_1, S_2 yield the same tree T . Let K^B be the set of reticulation vertices in $B-(w, v, P_1, P_2)$, where $B-(w, v, P_1, P_2)$ is the associated basic cyclic pair of $N-(w, v, P_1, P_2)$. Let J be the set of two-way tree-path reticulations with respect to w in $N-(w, v, P_1, P_2)$. By definition of $N-(w, v, P_1, P_2)$ and the construction of $B-(w, v, P_1, P_2)$, $K^B = J$. Let E_{K^B} be the set of reticulation edges of the reticulation vertices in K^B , and let E_J be the set of reticulation edges of the reticulation vertices in J . We see that $E_{K^B} = E_J$. Consider $S_1 \cap E_J$ and $S_2 \cap E_J$. Since $E_J = E_{K^B}$, we have $S_1 \cap E_J$ and $S_2 \cap E_J$ being two switchings of $B-(w, v, P_1, P_2)$. Since $\{e_1^v, e_2^v\} \subseteq E_J$, $e_1^v \in S_1$, and $e_2^v \in S_2$, we have $e_1^v \in S_1 \cap E_J$ and $e_2^v \in S_2 \cap E_J$. We will show that $S_1 \cap E_J$ and $S_2 \cap E_J$ yield the same tree, by using the two general switchings $S_1 \cup E_J$ and $S_2 \cup E_J$ of $N-(w, v, P_1, P_2)$. Let N_1 be the network yielded by $S_1 \cup E_J$, and let N_2 be the network yielded by $S_2 \cup E_J$. Take any reticulation vertex v_1 in N_1 then $v_1 \in J$, so v_1 is a two-way tree-path reticulation with respect to w in $N_1-(w, v, P_1, P_2)$. The reticulation vertex v_1 must also be a two-way tree-path reticulation with respect to w in $N_2-(w, v, P_1, P_2)$. Consider the tree $T_{v_1}^1$ below v_1 in $N_1-(w, v, P_1, P_2)$, and consider the tree $T_{v_1}^2$ below v_1 in $N_2-(w, v, P_1, P_2)$. We must have $T_{v_1}^1 = T_{v_1}^2$, since both $T_{v_1}^1$ and $T_{v_1}^2$ are on the same leaf set and they are subtrees of T . Coming back to $B-(w, v, P_1, P_2)$, replace every leaf ℓ_{v_1} by the tree T_{v_1} that is below v_1 in both N_1 and N_2 , call the result of doing this $B'-(w, v, P_1, P_2)$. We now see that $B'-(w, v, P_1, P_2)$ is a subnetwork of N_1 and N_2 ; hence applying $S_1 \cap E_J$ and $S_2 \cap E_J$ to $B'-(w, v, P_1, P_2)$ is the same as applying $S_1 \cap E_J$ to N_1 and applying $S_2 \cap E_J$ to N_2 . Therefore the result of applying $S_1 \cap E_J$ to $B'-(w, v, P_1, P_2)$ is a subtree, call it T_1 , of T on leaf set X' , and the result of applying $S_2 \cap E_J$ to $B'-(w, v, P_1, P_2)$ is a subtree, call it T_2 , of T on leaf set X' . Since $T = T$ and Lemma 4.1, $T_1 = T_2$. Thus $B'-(w, v, P_1, P_2)$ displays a tree twice, and $B-(w, v, P_1, P_2)$ must also display a tree twice.

Suppose that $B-(w, v, P_1, P_2)$ displays a tree twice. Then there exists two switchings S'_1, S'_2 of $B-(w, v, P_1, P_2)$ such that, without loss of generality, $e_1^v \in S'_1$ and $e_2^v \in S'_2$, and S'_1, S'_2 yield the

same tree T' . Let G be the general switching of $N-(w, v, P_1, P_2)$ such that $E_J \subseteq G$ and for any reticulation vertex outside of J exactly one of its reticulation edges is in G . Let N' be the network that is the result of applying G to $N-(w, v, P_1, P_2)$. Now, by construction, N' is a basic network. Take any reticulation vertex v_1 in N' then $v_1 \in J$, so that $v_1 \in K^B$. Now there is a tree below v_1 in N' , call it T_{v_1} . In $B-(w, v, P_1, P_2)$, for every such v_1 , replace the leaf ℓ_{v_1} with the tree T_{v_1} , call the result $B'-(w, v, P_1, P_2)$. We now see that $B'-(w, v, P_1, P_2)$ is a subnetwork of N' . Since $B-(w, v, P_1, P_2)$ displays a tree twice via S'_1, S'_2 , $B'-(w, v, P_1, P_2)$ displays a tree twice via S'_1, S'_2 ; hence N' displays a tree twice via S'_1, S'_2 . By Lemma 4.5, there exists two switchings S_1, S_2 of $N-(w, v, P_1, P_2)$ such that $e_1^v \in S_1$ and $e_2^v \in S_2$ and $N-(w, v, P_1, P_2)$ yields a tree twice via S_1 and S_2 . Therefore $N-(w, v, P_1, P_2)$ displays a tree twice. \square

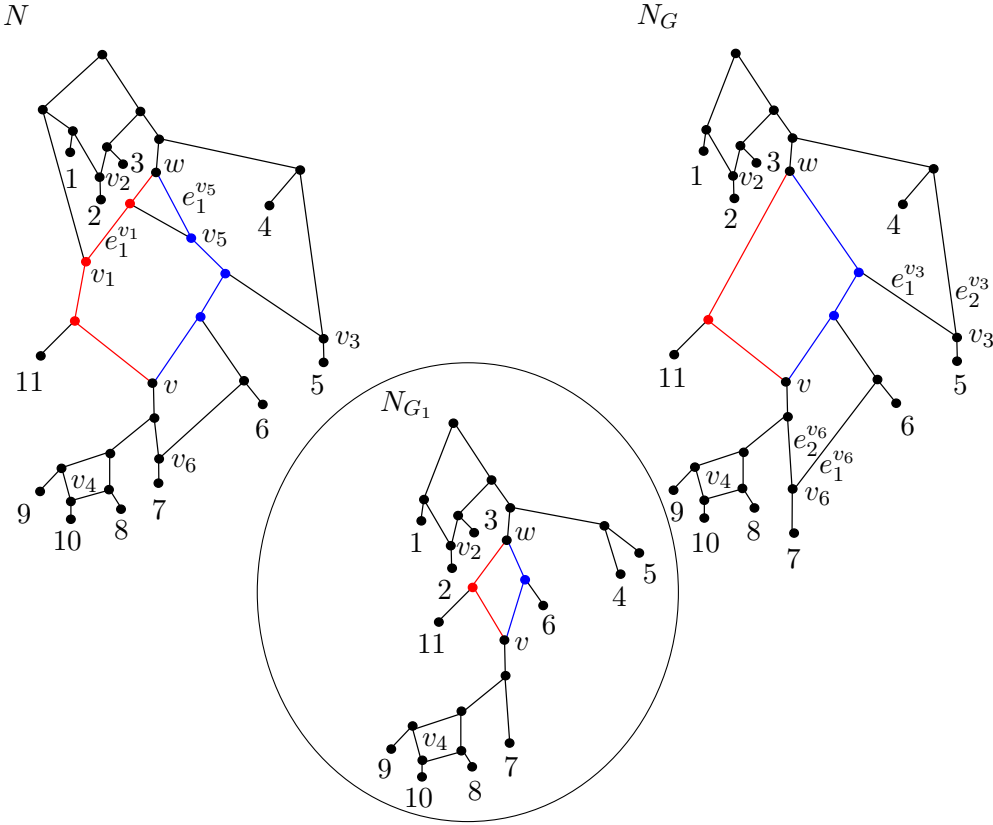
4.6.2 Processed cyclic pairs

It would not be efficient to have to check every maximal tree-path cyclic pair displayed by a network, because there are too many general switchings that result in maximal tree-path cyclic pairs. Hence we introduce a specific procedure that, when applied to a cyclic pair in a network, results in a maximal tree-path cyclic pair. The procedure applied to a cyclic pair consists of two general switchings, one applied after the other, with the result being a maximal tree-path cyclic pair, which we call the processed cyclic pair of that cyclic pair. The characterisation that is implicitly used in the efficient algorithm in deciding whether or not a network displays a tree twice is the following: A network without any avoidable reticulation vertices displays a tree twice if and only if there exists a cyclic pair whose processed cyclic pair displays a tree twice. This leads to the following corollary that is directly used in the efficient algorithm: A network without any avoidable reticulation vertices displays a tree twice if and only if there exists a cyclic pair whose processed cyclic pair has the property that its associated basic cyclic pair can not be reduced to a desirable configuration.

Definition 4.24. Let $N-(w, v, P_1, P_2)$ be a cyclic pair of the network N . The process applied to $N-(w, v, P_1, P_2)$ is the following procedure: Let D be the set of reticulation vertices on exactly one of P_1 or P_2 , so $v \notin D$ because v is on both P_1 and P_2 . For each $v_1 \in D$, let $e_1^{v_1}$ be the reticulation edge of v_1 that is on exactly one of P_1 or P_2 . Let G be the general switching of N such that, for each $v_1 \in D$, we have $\{e_1^{v_1}, e_2^{v_1}\} \cap G = \{e_1^{v_1}\}$, and, for every $v_2 \in K \setminus D$, we have $\{e_1^{v_2}, e_2^{v_2}\} \subseteq G$, where K is the set of reticulation vertices of N . Let N_G be the network that is the result of applying G to N . We see that $N_G-(w, v, P'_1, P'_2)$ is a tree-path cyclic pair of N_G , where P'_1 has the property that $V(P'_1) \subseteq V(P_1)$ and P'_2 has the property that $V(P'_2) \subseteq V(P_2)$. Let J be the set of tree-path reticulations with respect to w that are not two-way tree-path reticulations with respect to w , in

N_G . For each $v_3 \in J$ let $e_1^{v_3}$ be the reticulation edge of v_3 that is on the tree-path from w to v_3 . Hence $e_2^{v_3}$ is the reticulation edge of v_3 that is not on a tree-path from w to v_3 . Let G_1 be the general switching of N_G such that, for each $v_3 \in J$, we have $\{e_1^{v_3}, e_2^{v_3}\} \cap G_1 = \{e_2^{v_3}\}$, and, for every $v_4 \in K' \setminus J$, we have $\{e_1^{v_4}, e_2^{v_4}\} \subseteq G_1$, where K' is the set of reticulation vertices of N_G . Let N_{G_1} be the network that is the result of applying G_1 to N_G . We see that $N_{G_1}-(w, v, P_1'', P_2'')$ is a maximal tree-path cyclic pair of N_{G_1} , where P_1'' has the property that $V(P_1'') \subseteq V(P_1')$ and P_2'' has the property that $V(P_2'') \subseteq V(P_2')$. We call $N_{G_1}-(w, v, P_1'', P_2'')$ the *processed cyclic pair* of $N-(w, v, P_1, P_2)$.

Example 4.22. The network N has the cyclic pair $N-(w, v, P_1, P_2)$, where P_1 is coloured red and P_2 is coloured blue. The network N_G is obtained from N via the first part of the process applied to $N-(w, v, P_1, P_2)$. The network N_{G_1} has the processed cyclic pair of $N-(w, v, P_1, P_2)$ in it.



Theorem 4.2. Let N be a network without any avoidable reticulation vertices. The network N displays a tree twice if and only if there exists a cyclic pair $N-(w, v, P_1, P_2)$ whose processed cyclic pair $N_{G_1}-(w, v, P_1'', P_2'')$ displays a tree twice.

Proof. Suppose N displays a tree twice then, by Proposition 4.2, there exists a maximal tree-path cyclic pair $N'-(w, v, P_1', P_2')$, displayed by N , that displays a tree twice. Hence there exists two switchings S_1', S_2' of N' such that either $e_1^v \in S_1'$ and $e_2^v \in S_2'$ or $e_1^v \in S_2'$ and $e_2^v \in S_1'$, and S_1', S_2' yield

the same tree when they are applied to N' . The vertices w and v of N' exist in N . So in N there exists a cyclic pair $N-(w, v, P_1, P_2)$ such that $V(P'_1) \subseteq V(P_1)$ and $V(P'_2) \subseteq V(P_2)$. Now consider the processed cyclic pair $N_{G_1}-(w, v, P''_1, P''_2)$ of $N-(w, v, P_1, P_2)$. We see that $N'-(w, v, P'_1, P'_2)$ is displayed by N_{G_1} . Thus, by Lemma 4.5, N_{G_1} displays a tree twice. By Lemma 4.5, we see that the two switchings S_1, S_2 of N_{G_1} that yield the same tree have either $e_1^v \in S_1$ and $e_2^v \in S_2$ or $e_1^v \in S_2$ and $e_2^v \in S_1$. Therefore $N_{G_1}-(w, v, P''_1, P''_2)$ displays a tree twice.

Suppose there exists a cyclic pair $N-(w, v, P_1, P_2)$ whose processed cyclic pair $N_{G_1}-(w, v, P''_1, P''_2)$ displays a tree twice. Then, by definition, $N_{G_1}-(w, v, P''_1, P''_2)$ is displayed by N and displays a tree twice. Hence N_{G_1} displays a tree twice and is displayed by N . Therefore, by Lemma 4.5, N displays a tree twice. \square

Corollary 4.2. *Let N be a network without any avoidable reticulation vertices. The network N displays a tree twice if and only if there exists a cyclic pair $N-(w, v, P_1, P_2)$ whose processed cyclic pair $N_{G_1}-(w, v, P''_1, P''_2)$ has the following property: The associated basic cyclic pair of $N_{G_1}-(w, v, P''_1, P''_2)$ can not be reduced to a desirable configuration.*

Proof. Suppose N displays a tree twice then, by Theorem 4.2, there exists a cyclic pair $N-(w, v, P_1, P_2)$, whose processed cyclic pair $N_{G_1}-(w, v, P''_1, P''_2)$ displays a tree twice. By Proposition 4.3, the associated basic cyclic pair $B-(w, v, P''_1, P''_2)$ of $N_{G_1}-(w, v, P''_1, P''_2)$ displays a tree twice. By Corollary 4.1, $B-(w, v, P''_1, P''_2)$ can not be reduced to a desirable configuration.

Suppose there exists a cyclic pair $N-(w, v, P_1, P_2)$ whose processed cyclic pair $N_{G_1}-(w, v, P''_1, P''_2)$ has the following property: The associated basic cyclic pair $B-(w, v, P''_1, P''_2)$ of $N_{G_1}-(w, v, P''_1, P''_2)$ can not be reduced to a desirable configuration. Then, by Corollary 4.1, $B-(w, v, P''_1, P''_2)$ displays a tree twice. Hence, by Proposition 4.3, $N_{G_1}-(w, v, P''_1, P''_2)$ displays a tree twice. Therefore, by Theorem 4.2, N displays a tree twice. \square

Chapter 5

The Representation of a Basic Cyclic Pair

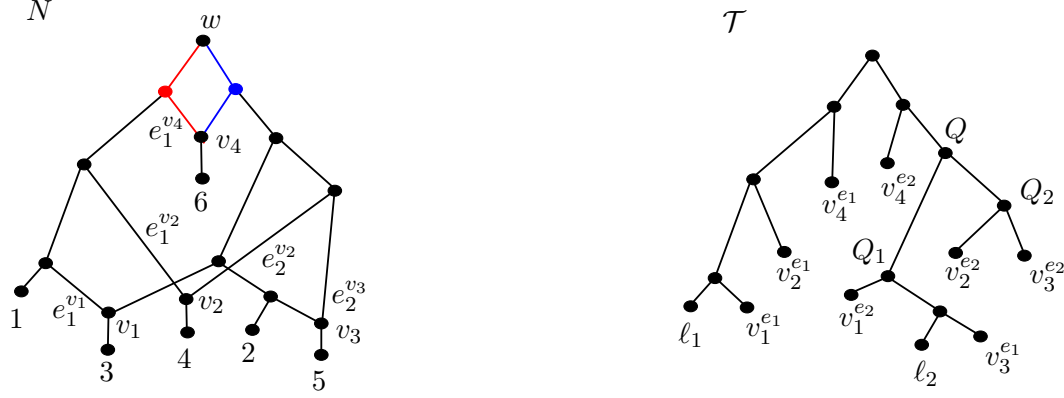
5.1 Introduction

In the last chapter we characterised a network without any avoidable reticulation vertices that displays a tree twice in terms of a substructure of the network. In this chapter we characterise the representation in terms of a substructure of the representation. With the benefit of hindsight, we see that the substructure of the representation has a deterministic characterisation. This deterministic characterisation lends itself to an efficient algorithm and provides insight into the harder problem of counting the number of trees displayed by a network. The substructure of the representation of a basic cyclic pair that we will be studying is a tree displayed by the representation that is itself a representation of a certain type of basic cyclic pair. This substructure has the following property: Each reticulation leaf is split between the two maximal subtrees of the representation. Every representation of a basic cyclic pair is a tree whose root has two children, and the trees whose roots are those two children are called the maximal subtrees of the representation. Having each reticulation leaf split between the two maximal subtrees means that if a reticulation leaf is in one maximal subtree then its partner is in the other maximal subtree.

Definition 5.1. Let \mathcal{T} be the representation of the basic cyclic pair $N-(w, v, P_1, P_2)$. Let Q be a subtree of \mathcal{T} such that the root of Q is not a leaf. Let the two children of the root of Q be x_1, x_2 . A *maximal subtree* of Q is a subtree of \mathcal{T} whose root is either x_1 or x_2 .

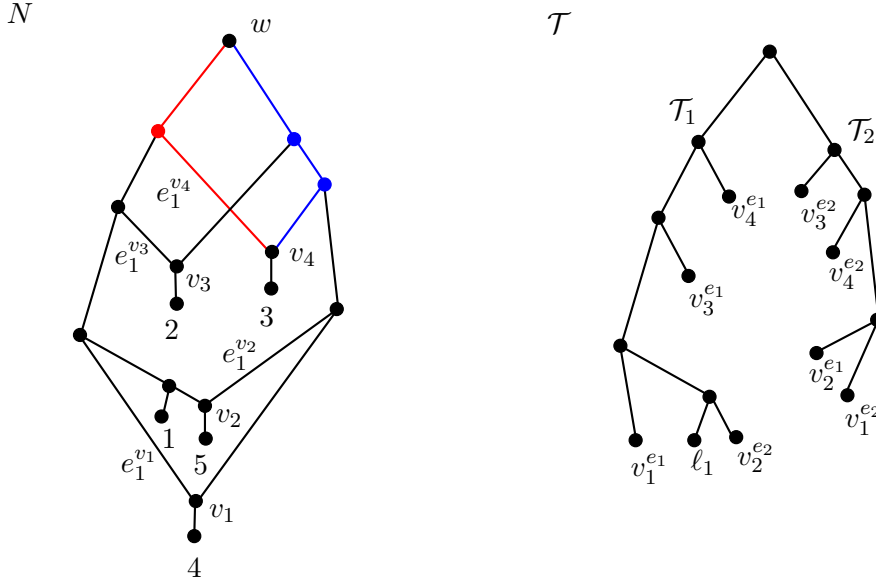
Example 5.1. Shown on the left is the basic cyclic pair $N-(w, v_4, P_1, P_2)$, where P_1 is in red whilst P_2 is in blue. Shown on the right is the representation \mathcal{T} of $N-(w, v_4, P_1, P_2)$. The subtree Q of \mathcal{T}

is labeled together with its two maximal subtrees Q_1, Q_2 .



Definition 5.2. Let \mathcal{T} be the representation of the basic cyclic pair $N-(w, v, P_1, P_2)$. Let \mathcal{T}_1 and \mathcal{T}_2 be \mathcal{T} 's two maximal subtrees. We say that \mathcal{T} is a *basic⁰ representation* when for each reticulation leaf $v_i^{e_m}$ in the leaf set of \mathcal{T}_1 , $v_i^{e_n}$, where $m \in \{1, 2\}$, $i \in \mathbb{N}$, and $n \in \{1, 2\} \setminus \{m\}$, is in the leaf set of \mathcal{T}_2 , and vice versa.

Example 5.2. Shown on the left is the basic cyclic pair $N-(w, v_4, P_1, P_2)$, where P_1 is in red whilst P_2 is in blue. Shown on the right is the *basic⁰ representation* \mathcal{T} of $N-(w, v_4, P_1, P_2)$. The tree shown in Example 5.1 is not a *basic⁰ representation*.



Definition 5.3. Let \mathcal{T} be the representation of the basic cyclic pair $N-(w, v, P_1, P_2)$. The tree \mathcal{T} displays a representation \mathcal{T}^* when there exists a general switching G of \mathcal{T} such that $\{v^{e_1}, v^{e_2}\} \subseteq G$ and $(\mathcal{T}|G)^L = \mathcal{T}^*$.

Theorem 5.1. Let \mathcal{T} be the representation of the basic cyclic pair $N-(w, v, P_1, P_2)$. The tree \mathcal{T} displays a tree twice if and only if there exists a *basic⁰ representation* \mathcal{T}^* that is displayed by \mathcal{T} and \mathcal{T}^* displays a tree twice relative to v .

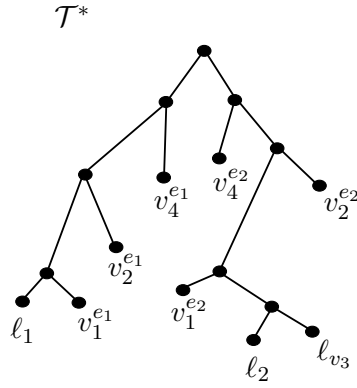
5.2 Characterising when the Representation of a Basic Cyclic Pair Displays a Tree Twice

Before we prove Theorem 5.1, two lemmas are needed. The two lemmas describe archetypal desirable configurations and the conditions that are needed in order to be able to reduce a representation to such an archetype. The first archetypal desirable configuration has two maximal subtrees which each have three reticulation leaves. In both maximal subtrees two of the reticulation leaves are partners whilst the other has its partner in the other maximal subtree. We will prove that this archetype is a desirable configuration, and we will show that if a representation has the property that both its maximal subtrees have two reticulation leaves that are partners then that representation never displays a tree twice. The second archetypal desirable configuration has one maximal subtree with a fixed leaf whilst the other maximal subtree has three reticulation leaves, two of which are partners whilst the other has its partner in the other maximal subtree. Similarly, we will prove that this archetype is a desirable configuration, and we will show that if a representation has the property that one of its maximal subtrees has a fixed leaf and there is a reticulation leaf with itself and its partner in the other maximal subtree then that representation never displays a tree twice.

Theorem 5.1 is actually a corollary of another theorem. This other theorem says that the basic^0 representation that is displayed is displayed using a general switching that preserves the most pairs of reticulation leaves, so that if a pair of reticulation leaves are not needed for a basic^0 representation to be displayed then that pair is kept. Hence the minimum number of changes are made to the original representation. Thus a basic^0 representation that is displayed in that way is said to be parsimoniously displayed.

Definition 5.4. Let \mathcal{T} be the representation of the basic cyclic pair $N-(w, v, P_1, P_2)$, and let \mathcal{T}_1 and \mathcal{T}_2 be \mathcal{T} 's two maximal subtrees. Let D be the set of reticulation leaves $v_j^{e_i}$, where $i \in \{1, 2\}$ and $j \in \mathbb{N}$, such that $v_j^{e_i} \in D$ if and only if either $v_j^{e_i} \in X_{\mathcal{T}_1}$ and $v_j^{e_n} \in X_{\mathcal{T}_2}$, where $n \in \{1, 2\} \setminus \{i\}$, or $v_j^{e_i} \in X_{\mathcal{T}_2}$ and $v_j^{e_n} \in X_{\mathcal{T}_1}$. We say that \mathcal{T} *parsimoniously displays* a basic^0 representation \mathcal{T}^* when there exists a general switching G such that $(\mathcal{T}|G)^L = \mathcal{T}^*$ and $D \subseteq G$.

Example 5.3. The tree \mathcal{T}^* shown is parsimoniously displayed by \mathcal{T} from Example 5.1 via the general switching $G = D \cup \{v_3^{e_1}, \ell_1, \ell_2\}$, where $D = \{v_1^{e_1}, v_1^{e_2}, v_2^{e_1}, v_2^{e_2}, v_4^{e_1}, v_4^{e_2}\}$. We see that \mathcal{T}^* is a basic^0 representation.



Lemma 5.1. *Let \mathcal{T} be the tree representing the basic cyclic pair $B-(w, v, P_1, P_2)$. Let \mathcal{T}_1 and \mathcal{T}_2 be the two maximal subtrees of \mathcal{T} . If there exists reticulation leaves v_1^{e1}, v_2^{e1} such that $\{v_1^{e1}, v_1^{e2}\} \subseteq X_{\mathcal{T}_1}$ and $\{v_2^{e1}, v_2^{e2}\} \subseteq X_{\mathcal{T}_2}$ then \mathcal{T} does not display a tree twice.*

Proof. Let Γ be the reduction of $B-(w, v, P_1, P_2)$ where $\Gamma = \{v_1^{e1}, v_1^{e2}, v_2^{e1}, v_2^{e2}, v^{e1}, v^{e2}\}$. In order for $\mathcal{T}|\Gamma$ to display a tree twice there has to be two switchings R_1, R_2 such that v^{e1}, v^{e2} are split between R_1 and R_2 , and $((\mathcal{T}|\Gamma)|R_1)^L = ((\mathcal{T}|\Gamma)|R_2)^L$. Without loss of generality, suppose that $v^{e1} \in X_{\mathcal{T}_1}$ and $v^{e2} \in X_{\mathcal{T}_2}$, where $X_{\mathcal{T}_1}, X_{\mathcal{T}_2}$ are the leaf sets of $\mathcal{T}_1, \mathcal{T}_2$ respectively. Consider $R_1 = \{v^{e1}, v_1^{e1}, v_2^{e1}\}$ and $R_2 = \{v^{e2}, v_1^{e2}, v_2^{e2}\}$ where $i, j, \alpha, \beta \in \{1, 2\}$. Since $\{v^{e1}, v_1^{e1}, v_1^{e2}\} \subseteq X_{\mathcal{T}_1}$ and $\{v^{e2}, v_2^{e1}, v_2^{e2}\} \subseteq X_{\mathcal{T}_2}$, $((\mathcal{T}|\Gamma)|R_1)^L$ has the triple $\ell_v \ell_{v_1} | \ell_{v_2}$ whilst $((\mathcal{T}|\Gamma)|R_2)^L$ has the triple $\ell_v \ell_{v_2} | \ell_{v_1}$, so, by Lemma 1.2, $((\mathcal{T}|\Gamma)|R_1)^L \neq ((\mathcal{T}|\Gamma)|R_2)^L$. Therefore $\mathcal{T}|\Gamma$ does not display a tree twice. Thus, by Corollary 4.1, $B-(w, v, P_1, P_2)$ does not display a tree twice. Therefore, by Theorem 4.1, \mathcal{T} does not display a tree twice. \square

Lemma 5.2. *Let \mathcal{T} be the tree representing the basic cyclic pair $B-(w, v, P_1, P_2)$. Let \mathcal{T}_1 and \mathcal{T}_2 be the two maximal subtrees of \mathcal{T} . If $C_{\mathcal{T}_2} \neq \emptyset$ and there exists a reticulation leaf v_1^{e1} such that $\{v_1^{e1}, v_1^{e2}\} \subseteq X_{\mathcal{T}_1}$ or $C_{\mathcal{T}_1} \neq \emptyset$ and there exists a reticulation leaf v_2^{e1} such that $\{v_2^{e1}, v_2^{e2}\} \subseteq X_{\mathcal{T}_2}$ then \mathcal{T} does not display a tree twice.*

Proof. Without loss of generality, suppose that $C_{\mathcal{T}_2} \neq \emptyset$ and there exists a reticulation leaf v_1^{e1} such that $\{v_1^{e1}, v_1^{e2}\} \subseteq X_{\mathcal{T}_1}$. Then there exists a fixed leaf $\ell \in X_{\mathcal{T}_2}$. Let Γ be the reduction of $B-(w, v, P_1, P_2)$ where $\Gamma = \{v_1^{e1}, v_1^{e2}, v^{e1}, v^{e2}, \ell\}$. In order for $\mathcal{T}|\Gamma$ to display a tree twice there has to be two switchings R_1, R_2 such that v^{e1}, v^{e2} are split between R_1 and R_2 , and $((\mathcal{T}|\Gamma)|R_1)^L = ((\mathcal{T}|\Gamma)|R_2)^L$. Without loss of generality, suppose that $v^{e1} \in X_{\mathcal{T}_1}$ and $v^{e2} \in X_{\mathcal{T}_2}$, where $X_{\mathcal{T}_1}, X_{\mathcal{T}_2}$ are the leaf sets of $\mathcal{T}_1, \mathcal{T}_2$ respectively. Consider $R_1 = \{v^{e1}, v_1^{e1}, \ell\}$ and $R_2 = \{v^{e2}, v_1^{e2}, \ell\}$ where $i, j \in \{1, 2\}$. Since $\{v^{e1}, v_1^{e1}, v_1^{e2}\} \subseteq X_{\mathcal{T}_1}$ and $\{v^{e2}, \ell\} \subseteq X_{\mathcal{T}_2}$, $((\mathcal{T}|\Gamma)|R_1)^L$ has the triple $\ell_v \ell_{v_1} | \ell$ whilst $((\mathcal{T}|\Gamma)|R_2)^L$ has the triple $\ell_v \ell | \ell_{v_1}$, so, by Lemma 1.2, $((\mathcal{T}|\Gamma)|R_1)^L \neq ((\mathcal{T}|\Gamma)|R_2)^L$. Therefore $\mathcal{T}|\Gamma$

does not display a tree twice. Thus, by Corollary 4.1, $B-(w, v, P_1, P_2)$ does not display a tree twice. Therefore, by Theorem 4.1, \mathcal{T} does not display a tree twice. \square

Theorem 5.2. *Let \mathcal{T} be the representation of the basic cyclic pair $N-(w, v, P_1, P_2)$. The tree \mathcal{T} displays a tree twice if and only if there exists a basic^0 representation \mathcal{T}^* that is parsimoniously displayed by \mathcal{T} and \mathcal{T}^* displays a tree twice relative to v .*

Proof. Suppose that \mathcal{T} displays a tree twice. Let \mathcal{T}_1 and \mathcal{T}_2 be the two maximal subtrees of \mathcal{T} . If \mathcal{T} is a basic^0 representation then we have trivially found a basic^0 representation displayed by \mathcal{T} , namely \mathcal{T} , that displays a tree twice. Suppose that \mathcal{T} is not a basic^0 representation then there exists $v_1^{e_1}$ such that either $\{v_1^{e_1}, v_1^{e_2}\} \subseteq X_{\mathcal{T}_1}$ or $\{v_1^{e_1}, v_1^{e_2}\} \subseteq X_{\mathcal{T}_2}$, where $X_{\mathcal{T}_1}, X_{\mathcal{T}_2}$ are the leaf sets of $\mathcal{T}_1, \mathcal{T}_2$ respectively. Without loss of generality, suppose that $\{v_1^{e_1}, v_1^{e_2}\} \subseteq X_{\mathcal{T}_1}$ then if there exists $v_2^{e_1}$ such that $\{v_2^{e_1}, v_2^{e_2}\} \subseteq X_{\mathcal{T}_2}$ then, by Lemma 5.1, \mathcal{T} does not display a tree twice - a contradiction; hence no such $v_2^{e_1}$ exists. Thus we can relabel all the reticulation leaves $v_j^{e_i}$, where $i \in \{1, 2\}$ and $j \in \mathbb{N}$, in \mathcal{T}_2 as $v_j^{e_2}$ and concurrently relabel $v_j^{e_n}$, where $n \in \{1, 2\} \setminus \{i\}$, in \mathcal{T}_1 as $v_j^{e_1}$. Let D be the set of reticulation leaves $v_j^{e_i}$ such that $v_j^{e_i} \in D$ if and only if either $v_j^{e_i} \in X_{\mathcal{T}_1}$ and $v_j^{e_n} \in X_{\mathcal{T}_2}$ or $v_j^{e_i} \in X_{\mathcal{T}_2}$ and $v_j^{e_n} \in X_{\mathcal{T}_1}$. It is the case that $H_{\mathcal{T}_2} \subseteq D$, and for all $v_j^{e_2} \in H_{\mathcal{T}_2}$ we have $\{v_j^{e_1}, v_j^{e_2}\} \subseteq D$. Thus $v_1^{e_1} \notin D$ and $v_1^{e_2} \notin D$.

Since \mathcal{T} displays a tree twice, there exists two switchings R_1, R_2 of \mathcal{T} such that, without loss of generality, $v^{e_1} \in R_1$ and $v^{e_2} \in R_2$, and $(\mathcal{T}|R_1)^L = (\mathcal{T}|R_2)^L$. Consider the set G_1 defined as $R_1 \cup D$, and consider the set G_2 defined as $R_2 \cup D$. Both G_1 and G_2 are general switchings of \mathcal{T} , and they both result in basic^0 representations $(\mathcal{T}|G_1)^L$ and $(\mathcal{T}|G_2)^L$. By construction, $(\mathcal{T}|G_1)^L$ and $(\mathcal{T}|G_2)^L$ are both parsimoniously displayed by \mathcal{T} . Observe that both $(\mathcal{T}|G_1)^L$ and $(\mathcal{T}|G_2)^L$ have \mathcal{T}_2 as one of their maximal subtrees. Let the other maximal subtree of $(\mathcal{T}|G_1)^L$ be called $\mathcal{T}_1^{G_1}$, and let the other maximal subtree of $(\mathcal{T}|G_2)^L$ be called $\mathcal{T}_1^{G_2}$. We want to prove that $(\mathcal{T}|G_1)^L$ displays a tree twice, but before we can do that we need a few more facts. First observe that $C_{\mathcal{T}_1^{G_1}}$ and $C_{\mathcal{T}_1^{G_2}}$ are both non-empty: They both contain ℓ_{v_1} . Next note that $C_{\mathcal{T}_2} = \emptyset$, because if it were not then, by Lemma 5.2, \mathcal{T} would not display a tree twice - a contradiction. We will use these facts to find out what $(\mathcal{T}|R_1)^L = (\mathcal{T}|R_2)^L$ implies.

We can split $R_1 \cap X_{(\mathcal{T}|G_1)^L}$ into two disjoint subsets, namely $R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_2}$ and $R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_1^{G_1}}$. Likewise we can split $R_2 \cap X_{(\mathcal{T}|G_2)^L}$ into two disjoint subsets, namely $R_2 \cap X_{(\mathcal{T}|G_2)^L} \cap X_{\mathcal{T}_2}$ and $R_2 \cap X_{(\mathcal{T}|G_2)^L} \cap X_{\mathcal{T}_1^{G_2}}$. Consider $(\mathcal{T}_2|R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_2})^L$ and $(\mathcal{T}_1^{G_1}|(R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_1^{G_1}}) \cup C_{\mathcal{T}_1^{G_1}})^L$, and consider $(\mathcal{T}_2|R_2 \cap X_{(\mathcal{T}|G_2)^L} \cap X_{\mathcal{T}_2})^L$ and $(\mathcal{T}_1^{G_2}|(R_2 \cap X_{(\mathcal{T}|G_2)^L} \cap X_{\mathcal{T}_1^{G_2}}) \cup C_{\mathcal{T}_1^{G_2}})^L$. Suppose that $R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_2} \neq \emptyset$ then the above

trees are the maximal subtrees of $(\mathcal{T}|R_1)^L$ and $(\mathcal{T}|R_2)^L$ respectively. Note that $R_2 \cap X_{(\mathcal{T}|G_2)^L} \cap X_{\mathcal{T}_2} \neq \emptyset$, because $v^{e_2} \in R_2$ and $v^{e_2} \in X_{\mathcal{T}_2}$, by definition, and $v^{e_2} \in X_{(\mathcal{T}|G_2)^L}$ as $\{v^{e_1}, v^{e_2}\} \subseteq D$. Now $(\mathcal{T}|R_1)^L = (\mathcal{T}|R_2)^L$ implies that either $(\mathcal{T}_2|R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_2})^L = (\mathcal{T}_2|R_2 \cap X_{(\mathcal{T}|G_2)^L} \cap X_{\mathcal{T}_2})^L$ and $(\mathcal{T}_1^{G_1}|(R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_1^{G_1}}) \cup C_{\mathcal{T}_1^{G_1}})^L = (\mathcal{T}_1^{G_2}|(R_2 \cap X_{(\mathcal{T}|G_2)^L} \cap X_{\mathcal{T}_1^{G_2}}) \cup C_{\mathcal{T}_1^{G_2}})^L$ or $(\mathcal{T}_2|R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_2})^L = (\mathcal{T}_1^{G_2}|(R_2 \cap X_{(\mathcal{T}|G_2)^L} \cap X_{\mathcal{T}_1^{G_2}}) \cup C_{\mathcal{T}_1^{G_2}})^L$ and $(\mathcal{T}_1^{G_1}|(R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_1^{G_1}}) \cup C_{\mathcal{T}_1^{G_1}})^L = (\mathcal{T}_2|R_2 \cap X_{(\mathcal{T}|G_2)^L} \cap X_{\mathcal{T}_2})^L$. We see that both possibilities are impossible, because in the former case ℓ_v is not in the leaf set of $(\mathcal{T}_2|R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_2})^L$ yet it is in the leaf set of $(\mathcal{T}_2|R_2 \cap X_{(\mathcal{T}|G_2)^L} \cap X_{\mathcal{T}_2})^L$. In the latter case ℓ_{v_1} is not in the leaf set of $(\mathcal{T}_2|R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_2})^L$ yet it is in the leaf set of $(\mathcal{T}_1^{G_2}|(R_2 \cap X_{(\mathcal{T}|G_2)^L} \cap X_{\mathcal{T}_1^{G_2}}) \cup C_{\mathcal{T}_1^{G_2}})^L$. Therefore we must have $R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_2} = \emptyset$. Thus $(\mathcal{T}|R_1)^L = (\mathcal{T}_1^{G_1}|(R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_1^{G_1}}) \cup C_{\mathcal{T}_1^{G_1}})^L$, and we still have $(\mathcal{T}_2|R_2 \cap X_{(\mathcal{T}|G_2)^L} \cap X_{\mathcal{T}_2})^L$ and $(\mathcal{T}_1^{G_2}|(R_2 \cap X_{(\mathcal{T}|G_2)^L} \cap X_{\mathcal{T}_1^{G_2}}) \cup C_{\mathcal{T}_1^{G_2}})^L$ being the maximal subtrees of $(\mathcal{T}|R_2)^L$. Note that $R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_2} = R_1 \cap X_{\mathcal{T}_2}$ and $R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_1^{G_1}} = R_1 \cap X_{\mathcal{T}_1^{G_1}}$. Since $R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_2} = \emptyset$, we have $R_1 \cap X_{\mathcal{T}_2} = \emptyset$. We claim that $(R_1 \cap X_{\mathcal{T}_1^{G_1}}) \cup C_{\mathcal{T}_1^{G_1}} = X_{\mathcal{T}_1^{G_1}}$. Take any $x \in (R_1 \cap X_{\mathcal{T}_1^{G_1}}) \cup C_{\mathcal{T}_1^{G_1}}$ then $x \in (R_1 \cap X_{\mathcal{T}_1^{G_1}})$ or $x \in C_{\mathcal{T}_1^{G_1}}$. In both cases we see that $x \in X_{\mathcal{T}_1^{G_1}}$, so $(R_1 \cap X_{\mathcal{T}_1^{G_1}}) \cup C_{\mathcal{T}_1^{G_1}} \subseteq X_{\mathcal{T}_1^{G_1}}$. Take any $x \in X_{\mathcal{T}_1^{G_1}}$ then either $x \in H_{\mathcal{T}_1^{G_1}}$ or $x \in C_{\mathcal{T}_1^{G_1}}$. The latter directly shows that $x \in (R_1 \cap X_{\mathcal{T}_1^{G_1}}) \cup C_{\mathcal{T}_1^{G_1}}$, so suppose that $x \in H_{\mathcal{T}_1^{G_1}}$. Then x is of form $v_j^{e_1}$, as $H_{\mathcal{T}_1^{G_1}} \subseteq H_{\mathcal{T}_1}$. Since R_1 is a switching, either $v_j^{e_1} \in R_1$ or $v_j^{e_2} \in R_1$. Note that $v_j^{e_2} \in X_{\mathcal{T}_2}$, so if $v_j^{e_2} \in R_1$ then $R_1 \cap X_{\mathcal{T}_2} \neq \emptyset$ - a contradiction. Thus $v_j^{e_1} \in R_1$; hence $x \in (R_1 \cap X_{\mathcal{T}_1^{G_1}})$. Therefore $(R_1 \cap X_{\mathcal{T}_1^{G_1}}) \cup C_{\mathcal{T}_1^{G_1}} = X_{\mathcal{T}_1^{G_1}}$. This implies that $(\mathcal{T}_1^{G_1}|(R_1 \cap X_{(\mathcal{T}|G_1)^L} \cap X_{\mathcal{T}_1^{G_1}}) \cup C_{\mathcal{T}_1^{G_1}})^L = (\mathcal{T}_1^{G_1}|X_{\mathcal{T}_1^{G_1}})^L$, and $(\mathcal{T}_1^{G_1}|X_{\mathcal{T}_1^{G_1}})^L = (\mathcal{T}_1^{G_1})^L$. Therefore $(\mathcal{T}|R_1)^L = (\mathcal{T}_1^{G_1})^L$.

Let the two maximal subtrees of $\mathcal{T}_1^{G_1}$ be $A_{\mathcal{T}_1^{G_1}}$ and $M_{\mathcal{T}_1^{G_1}}$. We see that, since $(\mathcal{T}|R_1)^L = (\mathcal{T}|R_2)^L$ and $(\mathcal{T}|R_1)^L = (\mathcal{T}_1^{G_1})^L$, we must have, without loss of generality, $(A_{\mathcal{T}_1^{G_1}})^L = (\mathcal{T}_2|R_2 \cap X_{(\mathcal{T}|G_2)^L} \cap X_{\mathcal{T}_2})^L$ and $(M_{\mathcal{T}_1^{G_1}})^L = (\mathcal{T}_1^{G_2}|(R_2 \cap X_{(\mathcal{T}|G_2)^L} \cap X_{\mathcal{T}_1^{G_2}}) \cup C_{\mathcal{T}_1^{G_2}})^L$. In order for these two equations to hold, we must have $v^{e_1} \in X_{A_{\mathcal{T}_1^{G_1}}}$, $\ell_{v_1} \in X_{M_{\mathcal{T}_1^{G_1}}}$, and $C_{A_{\mathcal{T}_1^{G_1}}} = \emptyset$. With these facts in hand, we will show that $(\mathcal{T}|G_1)^L$ displays a tree twice. Consider the two sets $R_{\mathcal{T}_1^{G_1}} \subseteq X_{\mathcal{T}_1^{G_1}}$ and $R_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$ defined as follows: We set $R_{\mathcal{T}_1^{G_1}} = X_{M_{\mathcal{T}_1^{G_1}}}$, and we define

$R_{\mathcal{T}_2}$ as having the property that $v_j^{e_2} \in R_{\mathcal{T}_2}$ if and only if $\ell_{v_j} \in X \left(A_{\mathcal{T}_1^{G_1}} \right)^L$. We claim that the set R_3 defined as $R_{\mathcal{T}_1^{G_1}} \cup R_{\mathcal{T}_2}$ is a switching of $(\mathcal{T}|G_1)^L$. Now $(\mathcal{T}|G_1)^L$ represents a basic cyclic pair, call it $B-(w', v, P'_1, P'_2)$. Take any reticulation vertex v_j in the set of reticulation vertices of $B-(w', v, P'_1, P'_2)$; we want to show that exactly one of $v_j^{e_1}, v_j^{e_2}$ is in R_3 . We have $v_j^{e_1} \in X_{(\mathcal{T}|G_1)^L}$, so $v_j^{e_1} \in X_{\mathcal{T}_1^{G_1}}$; hence either $v_j^{e_1} \in X_{M_{\mathcal{T}_1^{G_1}}}$ or $v_j^{e_1} \in X_{A_{\mathcal{T}_1^{G_1}}}$. If $v_j^{e_1} \in X_{M_{\mathcal{T}_1^{G_1}}}$ then $v_j^{e_1} \in R_{\mathcal{T}_1^{G_1}}$, so $v_j^{e_1} \in R_3$, and $v_j^{e_1} \notin X_{A_{\mathcal{T}_1^{G_1}}}$, so $\ell_{v_j} \notin X \left(A_{\mathcal{T}_1^{G_1}} \right)^L$; hence $v_j^{e_2} \notin R_{\mathcal{T}_2}$, so that $v_j^{e_2} \notin R_3$. If $v_j^{e_1} \in X_{A_{\mathcal{T}_1^{G_1}}}$ then $\ell_{v_j} \in X \left(A_{\mathcal{T}_1^{G_1}} \right)^L$, so that $v_j^{e_2} \in R_{\mathcal{T}_2}$; hence $v_j^{e_2} \in R_3$. We also see that $v_j^{e_1} \notin X_{M_{\mathcal{T}_1^{G_1}}}$, so that $v_j^{e_1} \notin R_{\mathcal{T}_1^{G_1}}$; thus $v_j^{e_1} \notin R_3$. Therefore R_3 is a switching of $(\mathcal{T}|G_1)^L$. Note that, since $v^{e_1} \in X_{A_{\mathcal{T}_1^{G_1}}}$, we have $v^{e_2} \in R_3$. Now $\left((\mathcal{T}|G_1)^L | R_3 \right)^L$ has maximal subtrees $\left(\mathcal{T}_1^{G_1} | R_{\mathcal{T}_1^{G_1}} \right)^L$ and $(\mathcal{T}_2 | R_{\mathcal{T}_2})^L$. Since $R_{\mathcal{T}_1^{G_1}} = X_{M_{\mathcal{T}_1^{G_1}}}$, we have $\left(\mathcal{T}_1^{G_1} | R_{\mathcal{T}_1^{G_1}} \right)^L = \left(\mathcal{T}_1^{G_1} | X_{M_{\mathcal{T}_1^{G_1}}} \right)^L$; by definition, $\mathcal{T}_1^{G_1} | X_{M_{\mathcal{T}_1^{G_1}}} = M_{\mathcal{T}_1^{G_1}}$. Therefore $\left(\mathcal{T}_1^{G_1} | R_{\mathcal{T}_1^{G_1}} \right)^L = \left(M_{\mathcal{T}_1^{G_1}} \right)^L$. We claim that $(\mathcal{T}_2 | R_{\mathcal{T}_2})^L = \left(A_{\mathcal{T}_1^{G_1}} \right)^L$. We will prove this claim, by showing that $(\mathcal{T}_2 | R_{\mathcal{T}_2})^L$ is equal to $(\mathcal{T}_2 | R_2 \cap X_{\mathcal{T}_2})^L$. In order to do that, we will show that $R_{\mathcal{T}_2} = R_2 \cap X_{\mathcal{T}_2}$. Recall that $\left(A_{\mathcal{T}_1^{G_1}} \right)^L = (\mathcal{T}_2 | R_2 \cap X_{\mathcal{T}_2})^L$, so $X \left(A_{\mathcal{T}_1^{G_1}} \right)^L = (R_2 \cap X_{\mathcal{T}_2})^L$. Take any $v_j^{e_2} \in R_{\mathcal{T}_2}$ then $\ell_{v_j} \in X \left(A_{\mathcal{T}_1^{G_1}} \right)^L$, which means that $\ell_{v_j} \in (R_2 \cap X_{\mathcal{T}_2})^L$. Since $v_j^{e_2} \in X_{\mathcal{T}_2}$, we must have $v_j^{e_2} \in R_2$, otherwise $\ell_{v_j} \notin (R_2 \cap X_{\mathcal{T}_2})^L$. Therefore $v_j^{e_2} \in R_2 \cap X_{\mathcal{T}_2}$. Take any $v_j^{e_m} \in R_2 \cap X_{\mathcal{T}_2}$ then $v_j^{e_m} \in X_{\mathcal{T}_2}$, so $v_j^{e_m} = v_j^{e_2}$. Thus $v_j^{e_2} \in R_2 \cap X_{\mathcal{T}_2}$. Now $\ell_{v_j} \in (R_2 \cap X_{\mathcal{T}_2})^L$; hence $\ell_{v_j} \in X \left(A_{\mathcal{T}_1^{G_1}} \right)^L$, so that $v_j^{e_2} \in R_{\mathcal{T}_2}$. Therefore $v_j^{e_m} \in R_{\mathcal{T}_2}$. Thus $R_{\mathcal{T}_2} = R_2 \cap X_{\mathcal{T}_2}$. Therefore $(\mathcal{T}_2 | R_{\mathcal{T}_2})^L = \left(A_{\mathcal{T}_1^{G_1}} \right)^L$. Recall that $(\mathcal{T}|R_1)^L = \left(\mathcal{T}_1^{G_1} \right)^L$, and the maximal subtrees of $\left(\mathcal{T}_1^{G_1} \right)^L$ are $\left(A_{\mathcal{T}_1^{G_1}} \right)^L$ and $\left(M_{\mathcal{T}_1^{G_1}} \right)^L$. Therefore $\left((\mathcal{T}|G_1)^L | R_3 \right)^L = (\mathcal{T}|R_1)^L$.

Consider the set R_4 defined as $(R_1 \cap D) \cup C_{\mathcal{T}_1^{G_1}}$. This is a switching of $(\mathcal{T}|G_1)^L$, because R_1 is a switching of \mathcal{T} and $G_1 = R_1 \cup D$. Since $v^{e_1} \in R_1$ and $\{v^{e_1}, v^{e_2}\} \subseteq D$, we have $v^{e_1} \in R_4$. We also see that $\left((\mathcal{T}|G_1)^L | R_4 \right)^L = (\mathcal{T}|R_1)^L$. Therefore $(\mathcal{T}|G_1)^L$ displays a tree twice via R_3 and R_4 .

Suppose there exists a basic⁰ representation \mathcal{T}^* that is parsimoniously displayed by \mathcal{T} and \mathcal{T}^* displays a tree twice. Then \mathcal{T}^* is displayed by \mathcal{T} , so there exists a general switching G of \mathcal{T} such that $\{v^{e_1}, v^{e_2}\} \subseteq G$ and $(\mathcal{T}|G)^L = \mathcal{T}^*$. Since \mathcal{T}^* displays a tree twice there exists two switchings R_1, R_2 of \mathcal{T}^* such that, without loss of generality, $v^{e_1} \in R_1$, $v^{e_2} \in R_2$, and $(\mathcal{T}^*|R_1)^L = (\mathcal{T}^*|R_2)^L$. Consider the two switchings R_3, R_4 of \mathcal{T} defined as having the following properties: The switching R_3 has $R_1 \subseteq R_3 \subseteq G$, and the switching R_4 has $R_2 \subseteq R_4 \subseteq G$. These properties do not invalidate

R_3 and R_4 being switchings of \mathcal{T} . Here is the proof: Take any reticulation leaf $v_j^{e_m}$ in \mathcal{T} , where $m \in \{1, 2\}$ and $j \in \mathbb{N}$. Either $v_j^{e_m}$ is in \mathcal{T}^* or it is not. Now exactly one of $v_j^{e_m}, v_j^{e_n}$, where $n \in \{1, 2\} \setminus \{m\}$, is in R_3 . Suppose that $v_j^{e_m}$ is in \mathcal{T}^* then $\{v_j^{e_1}, v_j^{e_2}\} \subseteq G$, and, since R_1 is a switching of \mathcal{T}^* , exactly one of $v_j^{e_1}, v_j^{e_2}$ is in R_1 . Without loss of generality, suppose that $v_j^{e_1} \in R_1$ then $v_j^{e_2} \notin R_1$, so $v_j^{e_1} \in R_3$ and $v_j^{e_2} \notin R_3$. Suppose that $v_j^{e_m}$ is not in \mathcal{T}^* then ℓ_{v_j} must be in \mathcal{T}^* , so that exactly one of $v_j^{e_1}, v_j^{e_2}$ is in G . Without loss of generality, suppose that $v_j^{e_1} \in G$ then $v_j^{e_2} \notin G$. We also must have $\{v_j^{e_1}, v_j^{e_2}\} \cap R_1 = \emptyset$. Therefore $v_j^{e_1} \in R_3$ and $v_j^{e_2} \notin R_3$. Likewise with R_4 . We also see that $(\mathcal{T}|R_3)^L = (\mathcal{T}^*|R_1)^L$ and $(\mathcal{T}|R_4)^L = (\mathcal{T}^*|R_2)^L$. Therefore $(\mathcal{T}|R_3)^L = (\mathcal{T}|R_4)^L$, and $v^{e_1} \in R_3$ and $v^{e_2} \in R_4$. Thus \mathcal{T} displays a tree twice. \square

5.3 Characterising when a Basic⁰ Representation Displays a Tree Twice

A basic⁰ representation has a characterisation of when it displays a tree twice in terms of subsets of the leaf set of the basic⁰ representation. In order to determine whether or not a basic⁰ representation displays a tree twice, subsets of its leaf set are used together with set operations such as intersection and union. If the subsets of the leaf set are known then it is easy to decide whether or not a basic⁰ representation displays a tree twice. Fortunately, a reasonable number of subsets of the basic⁰ representation's leaf set need to be known in order to quickly decide whether or not a basic⁰ representation displays a tree twice. By supposing that the basic⁰ representation displays a tree twice we can determine exactly what subsets of the basic⁰ representation's leaf set are needed.

Let \mathcal{T} be a basic⁰ representation of a basic cyclic pair $N-(w, v, P_1, P_2)$. Suppose that \mathcal{T} displays a tree twice. Then there exists two switchings R, W of \mathcal{T} such that $v^{e_1} \in R$ and $v^{e_2} \in W$ or $v^{e_2} \in R$ and $v^{e_1} \in W$ and $(\mathcal{T}|R)^L = (\mathcal{T}|W)^L$. Let \mathcal{T}_1 and \mathcal{T}_2 be \mathcal{T} 's two maximal subtrees, and let $X_{\mathcal{T}_1}$ and $X_{\mathcal{T}_2}$ be the leaf sets of \mathcal{T}_1 and \mathcal{T}_2 respectively. Let $R_{\mathcal{T}_1}$ be $R \cap X_{\mathcal{T}_1}$, and let $R_{\mathcal{T}_2}$ be $R \cap X_{\mathcal{T}_2}$. Likewise with $W_{\mathcal{T}_1}$ and $W_{\mathcal{T}_2}$. Without loss of generality, suppose that $v^{e_1} \in R$ and $v^{e_2} \in W$. Without loss of generality, suppose that $v^{e_1} \in X_{\mathcal{T}_1}$ and $v^{e_2} \in X_{\mathcal{T}_2}$. Then $v^{e_1} \in R_{\mathcal{T}_1}$ and $v^{e_2} \in W_{\mathcal{T}_2}$, so that $R_{\mathcal{T}_1} \neq \emptyset$ and $W_{\mathcal{T}_2} \neq \emptyset$. Let us now look at some cases in regards to $R_{\mathcal{T}_2}$ and $W_{\mathcal{T}_1}$.

5.3.1 Case 1: Both $R_{\mathcal{T}_2}$ and $W_{\mathcal{T}_1}$ are empty

The only way that both $R_{\mathcal{T}_2}$ and $W_{\mathcal{T}_1}$ can be empty is when there are no fixed leaves in \mathcal{T}_1 nor in \mathcal{T}_2 , so $C_{\mathcal{T}_1} = \emptyset$ and $C_{\mathcal{T}_2} = \emptyset$. Since \mathcal{T} is a basic⁰ representation, for any reticulation leaf $v_i^{e_m}$, where $m \in \{1, 2\}$ and $i \in \mathbb{N}$, in $X_{\mathcal{T}}$ either $v_i^{e_m} \in X_{\mathcal{T}_1}$ or $v_i^{e_m} \in X_{\mathcal{T}_2}$. Since R is a switching of \mathcal{T} , for any

reticulation leaf $v_i^{e_m}$ either $v_i^{e_m} \in R$ or $v_i^{e_n} \in R$, where $n \in \{1, 2\} \setminus \{m\}$. Therefore $R \cap X_{\mathcal{T}_1} = X_{\mathcal{T}_1}$, because $R \cap X_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$ and if we take any element in $X_{\mathcal{T}_1}$ then it must be a reticulation leaf, call it $v_i^{e_m}$, if $v_i^{e_m} \notin R$ then $v_i^{e_n} \in R$, but $v_i^{e_n} \in X_{\mathcal{T}_2}$, and this means that $R_{\mathcal{T}_2} \neq \emptyset$ - a contradiction. Therefore $R_{\mathcal{T}_1} = X_{\mathcal{T}_1}$ and $W_{\mathcal{T}_2} = X_{\mathcal{T}_2}$, so that $\mathcal{T}|R = \mathcal{T}|R_{\mathcal{T}_1} = \mathcal{T}|X_{\mathcal{T}_1} = \mathcal{T}_1$ and $\mathcal{T}|W = \mathcal{T}_2$. Since $(\mathcal{T}|R)^L = (\mathcal{T}|W)^L$, we must have $(\mathcal{T}_1)^L = (\mathcal{T}_2)^L$.

Proposition 5.1. *Let \mathcal{T} be a basic^0 representation of a basic cyclic pair $N-(w, v, P_1, P_2)$. Let \mathcal{T}_1 and \mathcal{T}_2 be \mathcal{T} 's two maximal subtrees. If $(\mathcal{T}_1)^L = (\mathcal{T}_2)^L$ then \mathcal{T} displays a tree twice.*

Proof. Since \mathcal{T} is a basic^0 representation, $X_{\mathcal{T}_1}$ and $X_{\mathcal{T}_2}$ are two switchings of \mathcal{T} such that v 's reticulation leaves are split between the two switchings. Since $\mathcal{T}|X_{\mathcal{T}_1} = \mathcal{T}_1$ and $\mathcal{T}|X_{\mathcal{T}_2} = \mathcal{T}_2$, we see that $(\mathcal{T}|X_{\mathcal{T}_1})^L = (\mathcal{T}|X_{\mathcal{T}_2})^L$ because $(\mathcal{T}_1)^L = (\mathcal{T}_2)^L$. \square

5.3.2 Case 2: Exactly one of $R_{\mathcal{T}_2}$ and $W_{\mathcal{T}_1}$ is empty

Suppose that $R_{\mathcal{T}_2} = \emptyset$ and $W_{\mathcal{T}_1} \neq \emptyset$ then $C_{\mathcal{T}_2} = \emptyset$ and $R_{\mathcal{T}_1} = X_{\mathcal{T}_1}$. Hence $(\mathcal{T}|R) = \mathcal{T}_1$. Since $(\mathcal{T}|R)^L = (\mathcal{T}|W)^L$, we have $(\mathcal{T}_1)^L = (\mathcal{T}|W)^L$. Let $A_{\mathcal{T}_1}$ and $M_{\mathcal{T}_1}$ be the two maximal subtrees of \mathcal{T}_1 , and let $X_{A_{\mathcal{T}_1}}$ and $X_{M_{\mathcal{T}_1}}$ be the leaf sets of $A_{\mathcal{T}_1}$ and $M_{\mathcal{T}_1}$ respectively. Since $v^{e_1} \in X_{\mathcal{T}_1}$, we must have either $v^{e_1} \in X_{A_{\mathcal{T}_1}}$ or $v^{e_1} \in X_{M_{\mathcal{T}_1}}$. Without loss of generality, suppose that $v^{e_1} \in X_{M_{\mathcal{T}_1}}$. Now $v^{e_2} \in W_{\mathcal{T}_2}$, so we must have $(\mathcal{T}_2|W_{\mathcal{T}_2})^L = (M_{\mathcal{T}_1})^L$, because $v^{e_2} \notin W_{\mathcal{T}_1}$ and so ℓ_v is not in the leaf set of $(\mathcal{T}_1|W_{\mathcal{T}_1})^L$ but it is in the leaf set of $(M_{\mathcal{T}_1})^L$. The only way that $(\mathcal{T}_2|W_{\mathcal{T}_2})^L$ can be equal to $(M_{\mathcal{T}_1})^L$ is when $(W_{\mathcal{T}_2})^L = (X_{M_{\mathcal{T}_1}})^L$, because the leaf set of $(\mathcal{T}_2|W_{\mathcal{T}_2})^L$ is $(W_{\mathcal{T}_2})^L$ and the leaf set of $(M_{\mathcal{T}_1})^L$ is $(X_{M_{\mathcal{T}_1}})^L$. Hence $(\mathcal{T}_2)^L | (X_{M_{\mathcal{T}_1}})^L = (M_{\mathcal{T}_1})^L$. Now we get a similar result for $W_{\mathcal{T}_1}$: Namely, $(\mathcal{T}_1)^L | (X_{A_{\mathcal{T}_1}})^L = (A_{\mathcal{T}_1})^L$, so $W_{\mathcal{T}_1} = X_{A_{\mathcal{T}_1}}$. Suppose that $W_{\mathcal{T}_1} = \emptyset$ and $R_{\mathcal{T}_2} \neq \emptyset$ then $C_{\mathcal{T}_1} = \emptyset$ and $W_{\mathcal{T}_2} = X_{\mathcal{T}_2}$. Hence $(\mathcal{T}|W) = \mathcal{T}_2$. Since $(\mathcal{T}|R)^L = (\mathcal{T}|W)^L$, we have $(\mathcal{T}|R)^L = (\mathcal{T}_2)^L$. Let $A_{\mathcal{T}_2}$ and $M_{\mathcal{T}_2}$ be the two maximal subtrees of \mathcal{T}_2 , and let $X_{A_{\mathcal{T}_2}}$ and $X_{M_{\mathcal{T}_2}}$ be the leaf sets of $A_{\mathcal{T}_2}$ and $M_{\mathcal{T}_2}$ respectively. Since $v^{e_2} \in X_{\mathcal{T}_2}$, we must have either $v^{e_2} \in X_{A_{\mathcal{T}_2}}$ or $v^{e_2} \in X_{M_{\mathcal{T}_2}}$. Without loss of generality, suppose that $v^{e_2} \in X_{M_{\mathcal{T}_2}}$. By a similar line of reasoning as before, we have $(\mathcal{T}_1)^L | (X_{M_{\mathcal{T}_2}})^L = (M_{\mathcal{T}_2})^L$ and $(\mathcal{T}_2)^L | (X_{A_{\mathcal{T}_2}})^L = (A_{\mathcal{T}_2})^L$.

Proposition 5.2. *Let \mathcal{T} be a basic^0 representation of a basic cyclic pair $N-(w, v, P_1, P_2)$. Let \mathcal{T}_1 and \mathcal{T}_2 be \mathcal{T} 's two maximal subtrees, and let $A_{\mathcal{T}_1}$ and $M_{\mathcal{T}_1}$ be the two maximal subtrees of \mathcal{T}_1 ; let $A_{\mathcal{T}_2}$ and $M_{\mathcal{T}_2}$ be the two maximal subtrees of \mathcal{T}_2 . Let $v^{e_1} \in X_{M_{\mathcal{T}_1}}$, and let $v^{e_2} \in X_{M_{\mathcal{T}_2}}$. If $(\mathcal{T}_2)^L | (X_{M_{\mathcal{T}_1}})^L = (M_{\mathcal{T}_1})^L$ and $C_{\mathcal{T}_2} = \emptyset$ or $(\mathcal{T}_1)^L | (X_{M_{\mathcal{T}_2}})^L = (M_{\mathcal{T}_2})^L$ and $C_{\mathcal{T}_1} = \emptyset$ then \mathcal{T} displays a tree twice.*

Proof. Suppose that $(\mathcal{T}_2)^L | (X_{M_{\mathcal{T}_1}})^L = (M_{\mathcal{T}_1})^L$ and $C_{\mathcal{T}_2} = \emptyset$ - we will find two switchings of \mathcal{T} that

yield the same tree. Since \mathcal{T} is a basic⁰ representation and $C_{\mathcal{T}_2} = \emptyset$, $R = X_{\mathcal{T}_1}$ is a switching of \mathcal{T} . Consider the following two sets $W_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$ and $W_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$ where $W_{\mathcal{T}_1} = X_{A_{\mathcal{T}_1}}$ and $W_{\mathcal{T}_2}$ has the property that $(W_{\mathcal{T}_2})^L = (X_{M_{\mathcal{T}_1}})^L$. We shall prove that W is a switching where W is defined as $W_{\mathcal{T}_1} \cup W_{\mathcal{T}_2}$. Take any reticulation leaf $v_i^{e_m}$ in $X_{\mathcal{T}}$, where $m \in \{1, 2\}$ and $i \in \mathbb{N}$, then either $v_i^{e_m} \in X_{\mathcal{T}_1}$ and $v_i^{e_n} \in X_{\mathcal{T}_2}$ or $v_i^{e_m} \in X_{\mathcal{T}_2}$ and $v_i^{e_n} \in X_{\mathcal{T}_1}$, where $n \in \{1, 2\} \setminus \{m\}$. Suppose that $v_i^{e_m} \in X_{\mathcal{T}_1}$ then $v_i^{e_n} \in X_{\mathcal{T}_2}$, as \mathcal{T} is a basic⁰ representation. Now either $v_i^{e_m} \in X_{A_{\mathcal{T}_1}}$ or $v_i^{e_m} \in X_{M_{\mathcal{T}_1}}$, suppose $v_i^{e_m} \in X_{A_{\mathcal{T}_1}}$ then $v_i^{e_m} \notin X_{M_{\mathcal{T}_1}}$; hence $v_i^{e_m} \in W$ and $\ell_{v_i} \notin (X_{M_{\mathcal{T}_1}})^L = (W_{\mathcal{T}_2})^L$; thus $v_i^{e_n} \notin W$. Suppose that $v_i^{e_m} \in X_{M_{\mathcal{T}_1}}$ then $v_i^{e_m} \notin X_{A_{\mathcal{T}_1}}$, so $v_i^{e_m} \notin W$, and $\ell_{v_i} \in (X_{M_{\mathcal{T}_1}})^L$; hence $v_i^{e_n} \in W$. A similar case is when $v_i^{e_m} \in X_{\mathcal{T}_2}$ and $v_i^{e_n} \in X_{\mathcal{T}_1}$. We also see that $C_{\mathcal{T}_1} \subseteq X_{A_{\mathcal{T}_1}}$, so that $C_{\mathcal{T}_1} \subseteq W$. Therefore W is a switching of \mathcal{T} . Since $v^{e_1} \in X_{M_{\mathcal{T}_1}}$, we have $v^{e_1} \in R$, and since $v^{e_2} \in X_{M_{\mathcal{T}_2}}$, we have $v^{e_2} \in W$. So that v 's reticulation leaves are split between R and W . Now we need to prove that $(\mathcal{T}|R)^L = (\mathcal{T}|W)^L$. Since $R = X_{\mathcal{T}_1}$, we get $(\mathcal{T}|R)^L = (\mathcal{T}_1)^L$, so we need to prove that $(\mathcal{T}_1)^L = (\mathcal{T}|W)^L$. Two trees are equal when their two maximal subtrees are equal, and \mathcal{T}_1 's two maximal subtrees are $A_{\mathcal{T}_1}$ and $M_{\mathcal{T}_1}$. Now the two maximal subtrees of $(\mathcal{T}|W)$ are $\mathcal{T}_1|W_{\mathcal{T}_1}$ and $\mathcal{T}_2|W_{\mathcal{T}_2}$. Since $W_{\mathcal{T}_1} = X_{A_{\mathcal{T}_1}}$, we get $\mathcal{T}_1|W_{\mathcal{T}_1} = A_{\mathcal{T}_1}$, so $(\mathcal{T}_1|W_{\mathcal{T}_1})^L = (A_{\mathcal{T}_1})^L$; hence we only need to show that $(\mathcal{T}_2|W_{\mathcal{T}_2})^L = (M_{\mathcal{T}_1})^L$. Observe that $(\mathcal{T}_2|W_{\mathcal{T}_2})^L = (\mathcal{T}_2)^L | (W_{\mathcal{T}_2})^L$, and $(W_{\mathcal{T}_2})^L = (X_{M_{\mathcal{T}_1}})^L$. Therefore $(\mathcal{T}_2|W_{\mathcal{T}_2})^L = (\mathcal{T}_2)^L | (X_{M_{\mathcal{T}_1}})^L$, and, by hypothesis, $(\mathcal{T}_2)^L | (X_{M_{\mathcal{T}_1}})^L = (M_{\mathcal{T}_1})^L$. Hence $(\mathcal{T}_2|W_{\mathcal{T}_2})^L = (M_{\mathcal{T}_1})^L$. Thus \mathcal{T} displays a tree twice. If $(\mathcal{T}_1)^L | (X_{M_{\mathcal{T}_2}})^L = (M_{\mathcal{T}_2})^L$ and $C_{\mathcal{T}_1} = \emptyset$ then \mathcal{T} displays a tree twice in a similar way as above. \square

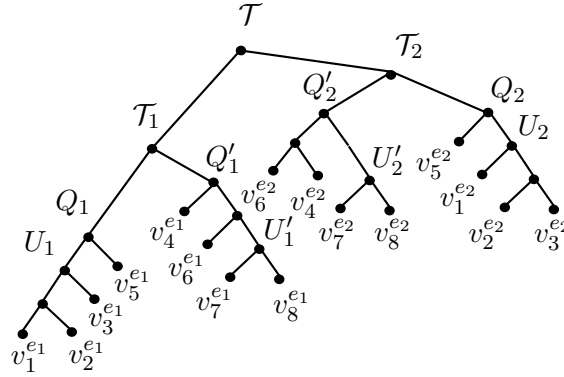
5.3.3 The final case

The final case is when both $R_{\mathcal{T}_2}$ and $W_{\mathcal{T}_1}$ are non-empty, and since v^{e_1} is in $R_{\mathcal{T}_1}$ and v^{e_2} is in $W_{\mathcal{T}_2}$, we immediately see that there is only one way in which $(\mathcal{T}|R)^L$ can be equal to $(\mathcal{T}|W)^L$. This implies that there are no fixed clusters in \mathcal{T} , and if we are going to get anything interesting then \mathcal{T} 's maximal subtrees $\mathcal{T}_1, \mathcal{T}_2$ have to have $(\mathcal{T}_1)^L \neq (\mathcal{T}_2)^L$. The properties of $\mathcal{T}_1, \mathcal{T}_2$, namely $(\mathcal{T}_1)^L \neq (\mathcal{T}_2)^L$ and $(X_{\mathcal{T}_1})^L = (X_{\mathcal{T}_2})^L$, turn out to be very important, so that the unordered pair $(\mathcal{T}_1, \mathcal{T}_2)$ is given the name of major tree pair. Formally, a major tree pair of \mathcal{T} is an unordered pair (Q_i, Q_j) , for $i \in \{1, 2\}$ and $j \in \{1, 2\} \setminus \{i\}$, where Q_i is a subtree of \mathcal{T}_i and Q_j is a subtree of \mathcal{T}_j having the following properties: $(Q_1)^L \neq (Q_2)^L$ and $(X_{Q_1})^L = (X_{Q_2})^L$. More generally we say that an unordered pair (Q_i, Q_j) is a tree pair when Q_i is a subtree of \mathcal{T}_i and Q_j is a subtree of \mathcal{T}_j having the following property: $(X_{Q_1})^L = (X_{Q_2})^L$. It really gets interesting when we have a major tree pair, say $(\mathcal{T}_1, \mathcal{T}_2)$, where none of the maximal subtrees of \mathcal{T}_1 and \mathcal{T}_2 make a major tree pair; we call such a major tree pair of \mathcal{T} a sterile major tree pair of \mathcal{T} . It is these sterile major tree pairs of \mathcal{T}

that characterise the final case.

Definition 5.5. Let \mathcal{T} be the representation of the basic cyclic pair $N-(w, v, P_1, P_2)$. Let \mathcal{T}_1 and \mathcal{T}_2 be \mathcal{T} 's two maximal subtrees. A *tree pair* of \mathcal{T} is an unordered pair (Q_i, Q_j) , for $i \in \{1, 2\}$ and $j \in \{1, 2\} \setminus \{i\}$, where Q_i is a subtree of \mathcal{T}_i and Q_j is a subtree of \mathcal{T}_j such that $(X_{Q_1})^L = (X_{Q_2})^L$, where X_{Q_1} is the leaf set of Q_1 and X_{Q_2} is the leaf set of Q_2 .

Example 5.4. The tree \mathcal{T} shown has many tree pairs. They are (U_1, U_2) , (Q_1, Q_2) , (U'_1, U'_2) , (Q'_1, Q'_2) , $(\mathcal{T}_1, \mathcal{T}_2)$, and (Q_1^i, Q_2^i) for $i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$, where Q_1^i is the subtree of \mathcal{T}_1 consisting of the leaf $v_i^{e_1}$ and Q_2^i is the subtree of \mathcal{T}_2 consisting of the leaf $v_i^{e_2}$.



Definition 5.6. Let \mathcal{T} be the representation of the basic cyclic pair $N-(w, v, P_1, P_2)$. A *trivial tree pair* of \mathcal{T} is a tree pair (Q_i, Q_j) , where $i \in \{1, 2\}$ and $j \in \{1, 2\} \setminus \{i\}$, of \mathcal{T} such that $(Q_1)^L = (Q_2)^L$.

Example 5.5. The trivial tree pairs from Example 5.4 are: (U'_1, U'_2) and (Q_1^i, Q_2^i) for $i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$.

Definition 5.7. Let \mathcal{T} be the representation of the basic cyclic pair $N-(w, v, P_1, P_2)$. A tree pair of \mathcal{T} (Q_i, Q_j) , where $i \in \{1, 2\}$ and $j \in \{1, 2\} \setminus \{i\}$, is a *major tree pair* of \mathcal{T} when $(Q_1)^L \neq (Q_2)^L$.

Example 5.6. The major tree pairs from Example 5.4 are: (U_1, U_2) , (Q_1, Q_2) , (Q'_1, Q'_2) , and $(\mathcal{T}_1, \mathcal{T}_2)$.

Definition 5.8. Let \mathcal{T} be the representation of the basic cyclic pair $N-(w, v, P_1, P_2)$. A major tree pair of \mathcal{T} (Q_i, Q_j) , where $i \in \{1, 2\}$ and $j \in \{1, 2\} \setminus \{i\}$, is a *sterile major tree pair* of \mathcal{T} when there is no other major tree pair (Q'_i, Q'_j) such that Q'_i is a maximal subtree of Q_i and Q'_j is a maximal subtree of Q_j .

Example 5.7. The sterile major tree pairs from Example 5.4 are: (Q'_1, Q'_2) and (U_1, U_2) .

Proposition 5.3. Let \mathcal{T} be the basic⁰ representation of the basic cyclic pair $N-(w, v, P_1, P_2)$, and let $C_{\mathcal{T}} = \emptyset$. Let \mathcal{T}_1 and \mathcal{T}_2 be \mathcal{T} 's two maximal subtrees. If $(\mathcal{T}_1)^L \neq (\mathcal{T}_2)^L$ then there exists a sterile major tree pair of \mathcal{T} .

Proof. Since \mathcal{T} is a basic^0 representation and $C_{\mathcal{T}} = \emptyset$, we have $(X_{\mathcal{T}_1})^L = (X_{\mathcal{T}_2})^L$; hence $(\mathcal{T}_1, \mathcal{T}_2)$ is a major tree pair of \mathcal{T} . Let \mathcal{D} be the set of all major tree pairs of \mathcal{T} . We can order the major tree pairs in \mathcal{D} by descendance as follows: A major tree pair (Q'_1, Q'_2) is a descendant of another major tree pair (Q_1, Q_2) when Q'_1 is a maximal subtree of Q_1 and Q'_2 is a maximal subtree of Q_2 . Since \mathcal{D} is finite, there exists a major tree pair $(Q'_1, Q'_2) \in \mathcal{D}$ that has no descendants in \mathcal{D} . By construction, (Q'_1, Q'_2) is a sterile major tree pair of \mathcal{T} . \square

With the existence of sterile major tree pairs established, we now return to the final case. In the final case the basic^0 representation displays a tree twice and both $R_{\mathcal{T}_2}$ and $W_{\mathcal{T}_1}$ are non-empty. Recall that $v^{e_1} \in R_{\mathcal{T}_1}$ and $v^{e_2} \in W_{\mathcal{T}_2}$, so that $v^{e_2} \notin R_{\mathcal{T}_2}$ and $v^{e_1} \notin W_{\mathcal{T}_1}$; hence $(\mathcal{T}_1|R_{\mathcal{T}_1})^L \neq (\mathcal{T}_1|W_{\mathcal{T}_1})^L$ and $(\mathcal{T}_2|R_{\mathcal{T}_2})^L \neq (\mathcal{T}_2|W_{\mathcal{T}_2})^L$, because these trees are on different leaf sets. The only way that $(\mathcal{T}|R)^L$ can be equal to $(\mathcal{T}|W)^L$, in this case, is when $(\mathcal{T}_1|R_{\mathcal{T}_1})^L = (\mathcal{T}_2|W_{\mathcal{T}_2})^L$ and $(\mathcal{T}_2|R_{\mathcal{T}_2})^L = (\mathcal{T}_1|W_{\mathcal{T}_1})^L$. Therefore $(R_{\mathcal{T}_1})^L = (W_{\mathcal{T}_2})^L$ and $(R_{\mathcal{T}_2})^L = (W_{\mathcal{T}_1})^L$. Observe that $(R_{\mathcal{T}_1})^L = (W_{\mathcal{T}_2})^L$ implies that $C_{\mathcal{T}_1} = \emptyset$ and $C_{\mathcal{T}_2} = \emptyset$, so that $(X_{\mathcal{T}_1})^L = (X_{\mathcal{T}_2})^L$; thus $(R_{\mathcal{T}_2})^L = (X_{\mathcal{T}_1})^L \setminus (R_{\mathcal{T}_1})^L$. Hence if we can completely determine what $(R_{\mathcal{T}_1})^L$ is then we can completely determine what $(R_{\mathcal{T}_2})^L$ is, and then we can completely determine what $(W_{\mathcal{T}_1})^L$ and $(W_{\mathcal{T}_2})^L$ are. Now if $(\mathcal{T}_1)^L = (\mathcal{T}_2)^L$ then \mathcal{T} displays a tree twice, so suppose that $(\mathcal{T}_1)^L \neq (\mathcal{T}_2)^L$ then $(\mathcal{T}_1, \mathcal{T}_2)$ is a major tree pair of \mathcal{T} .

By Proposition 5.3, there exists a sterile major tree pair of \mathcal{T} . Let (Q_1, Q_2) be any sterile major tree pair of \mathcal{T} . Let A_{Q_1} and M_{Q_1} be the two maximal subtrees of Q_1 , and let A_{Q_2} and M_{Q_2} be the two maximal subtrees of Q_2 . We will show that none of the maximal subtrees of Q_1 make a tree pair with any subtree of Q_2 .

Lemma 5.3. *Let (Q_1, Q_2) be a major tree pair of a basic^0 representation \mathcal{T} . Let A_{Q_1} and M_{Q_1} be the two maximal subtrees of Q_1 , and let A_{Q_2} and M_{Q_2} be the two maximal subtrees of Q_2 . If (Q_1, Q_2) is a sterile major tree pair of \mathcal{T} then $(X_{M_{Q_1}})^L \neq (X_{M_{Q_2}})^L$ and $(X_{M_{Q_1}})^L \neq (X_{A_{Q_2}})^L$. Moreover, $(X_{A_{Q_1}})^L \neq (X_{M_{Q_2}})^L$ and $(X_{A_{Q_1}})^L \neq (X_{A_{Q_2}})^L$.*

Proof. (Proof by contrapositive.) Without loss of generality, suppose that $(X_{M_{Q_1}})^L = (X_{M_{Q_2}})^L$. Since (Q_1, Q_2) is a tree pair of \mathcal{T} , $(X_{Q_1})^L = (X_{Q_2})^L$; hence $(X_{M_{Q_1}})^L \cup (X_{A_{Q_1}})^L = (X_{Q_1})^L$ and $(X_{Q_1})^L \setminus (X_{M_{Q_1}})^L = (X_{A_{Q_1}})^L$ (likewise with $(X_{M_{Q_2}})^L$ and $(X_{A_{Q_2}})^L$). Thus $(X_{Q_1})^L \setminus (X_{M_{Q_1}})^L = (X_{Q_1})^L \setminus (X_{M_{Q_2}})^L$, which implies that $(X_{Q_1})^L \setminus (X_{M_{Q_1}})^L = (X_{Q_2})^L \setminus (X_{M_{Q_2}})^L$. Therefore $(X_{A_{Q_1}})^L = (X_{A_{Q_2}})^L$. Now we can not have $(M_{Q_1})^L = (M_{Q_2})^L$ and $(A_{Q_1})^L = (A_{Q_2})^L$, as this implies that $(Q_1)^L = (Q_2)^L$ - a contradiction of the fact that (Q_1, Q_2) is a major tree pair of \mathcal{T} . Hence $(M_{Q_1})^L \neq (M_{Q_2})^L$ or $(A_{Q_1})^L \neq (A_{Q_2})^L$, and this implies that (M_{Q_1}, M_{Q_2}) or (A_{Q_1}, A_{Q_2}) is a major tree pair of \mathcal{T} . Therefore (Q_1, Q_2) is not a sterile major tree pair of \mathcal{T} . \square

Lemma 5.4. Let \mathcal{T} be the basic^0 representation of the basic cyclic pair $N-(w, v, P_1, P_2)$, and let $C_{\mathcal{T}} = \emptyset$. Let \mathcal{T}_1 and \mathcal{T}_2 be \mathcal{T} 's two maximal subtrees. Let R be a switching of the basic^0 representation \mathcal{T} of the basic cyclic pair, and let W be a switching of the basic^0 representation \mathcal{T} of the basic cyclic pair. Let $R_{\mathcal{T}_1}$ and $R_{\mathcal{T}_2}$ be subsets of R such that $R_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $R_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, and $R_{\mathcal{T}_1} \cup R_{\mathcal{T}_2} = R$, where \mathcal{T}_1 and \mathcal{T}_2 are \mathcal{T} 's two maximal subtrees, and let $W_{\mathcal{T}_1}$ and $W_{\mathcal{T}_2}$ be subsets of W such that $W_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $W_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, and $W_{\mathcal{T}_1} \cup W_{\mathcal{T}_2} = W$. If $(\mathcal{T}_1|R_{\mathcal{T}_1})^L = (\mathcal{T}_2|W_{\mathcal{T}_2})^L$ and $(\mathcal{T}_2|R_{\mathcal{T}_2})^L = (\mathcal{T}_1|W_{\mathcal{T}_1})^L$ then $(\mathcal{T}_1)^L|(R_{\mathcal{T}_1})^L = (\mathcal{T}_2)^L|(R_{\mathcal{T}_1})^L$ and $(\mathcal{T}_2)^L|(R_{\mathcal{T}_2})^L = (\mathcal{T}_1)^L|(R_{\mathcal{T}_2})^L$.

Proof. We know that $(\mathcal{T}_1|R_{\mathcal{T}_1})^L = (\mathcal{T}_1)^L|(R_{\mathcal{T}_1})^L$, so that $(\mathcal{T}_1|R_{\mathcal{T}_1})^L = (\mathcal{T}_2|W_{\mathcal{T}_2})^L$ implies $(\mathcal{T}_1)^L|(R_{\mathcal{T}_1})^L = (\mathcal{T}_2)^L|(W_{\mathcal{T}_2})^L$. Now we must have $(R_{\mathcal{T}_1})^L = (W_{\mathcal{T}_2})^L$; hence $(\mathcal{T}_1)^L|(R_{\mathcal{T}_1})^L = (\mathcal{T}_2)^L|(R_{\mathcal{T}_1})^L$. Similarly, $(\mathcal{T}_2|R_{\mathcal{T}_2})^L = (\mathcal{T}_1|W_{\mathcal{T}_1})^L$ implies that $(\mathcal{T}_2)^L|(R_{\mathcal{T}_2})^L = (\mathcal{T}_1)^L|(R_{\mathcal{T}_2})^L$. \square

Lemma 5.5. Let \mathcal{T} be the basic^0 representation of the basic cyclic pair $N-(w, v, P_1, P_2)$, and let $C_{\mathcal{T}} = \emptyset$. Let \mathcal{T}_1 and \mathcal{T}_2 be \mathcal{T} 's two maximal subtrees. Let R be a switching of the basic^0 representation \mathcal{T} of the basic cyclic pair, and let W be a switching of the basic^0 representation \mathcal{T} of the basic cyclic pair. Let $R_{\mathcal{T}_1}$ and $R_{\mathcal{T}_2}$ be subsets of R such that $R_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $R_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, and $R_{\mathcal{T}_1} \cup R_{\mathcal{T}_2} = R$, where \mathcal{T}_1 and \mathcal{T}_2 are \mathcal{T} 's two maximal subtrees. If $(\mathcal{T}_1)^L|(R_{\mathcal{T}_1})^L = (\mathcal{T}_2)^L|(R_{\mathcal{T}_1})^L$ and $(\mathcal{T}_2)^L|(R_{\mathcal{T}_2})^L = (\mathcal{T}_1)^L|(R_{\mathcal{T}_2})^L$ then $(Q_1)^L|(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = (Q_2)^L|(R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$ and $(Q_2)^L|(R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L = (Q_1)^L|(R_{\mathcal{T}_2})^L \cap (X_{Q_1})^L$, where (Q_1, Q_2) is a tree pair of \mathcal{T} .

Proof. (Contrapositive.) Suppose that $(Q_1)^L|(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L \neq (Q_2)^L|(R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$ then we have a subtree of $(\mathcal{T}_1)^L|(R_{\mathcal{T}_1})^L$ on leaf set $(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L$ that is not equal to a subtree of $(\mathcal{T}_2)^L|(R_{\mathcal{T}_1})^L$ on leaf set $(R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L = (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L$, as $(X_{Q_1})^L = (X_{Q_2})^L$. Therefore, by Lemma 4.1, $(\mathcal{T}_1)^L|(R_{\mathcal{T}_1})^L \neq (\mathcal{T}_2)^L|(R_{\mathcal{T}_1})^L$. Likewise if $(Q_2)^L|(R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L \neq (Q_1)^L|(R_{\mathcal{T}_2})^L \cap (X_{Q_1})^L$. \square

Now we need to make the following assumption: We assume that for every tree pair (Q_1, Q_2) we have $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$, $(X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$, $(X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$, and $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$, where A_{Q_1} and M_{Q_1} are the maximal subtrees of Q_1 , and A_{Q_2} and M_{Q_2} are the maximal subtrees of Q_2 ; we make that assumption so that the characterisation is easier to digest. Given the above assumption, we now set out to prove that $(Q_1)^L|(R_{\mathcal{T}_i})^L \cap (X_{Q_1})^L = (Q_2)^L|(R_{\mathcal{T}_i})^L \cap (X_{Q_2})^L$ implies that either $(M_{Q_1})^L|(R_{\mathcal{T}_i})^L \cap (X_{M_{Q_1}})^L = (M_{Q_2})^L|(R_{\mathcal{T}_i})^L \cap (X_{M_{Q_2}})^L$ and $(A_{Q_1})^L|(R_{\mathcal{T}_i})^L \cap (X_{A_{Q_1}})^L = (A_{Q_2})^L|(R_{\mathcal{T}_i})^L \cap (X_{A_{Q_2}})^L$ or $(M_{Q_1})^L|(R_{\mathcal{T}_i})^L \cap (X_{M_{Q_1}})^L = (A_{Q_2})^L|(R_{\mathcal{T}_i})^L \cap (X_{A_{Q_2}})^L$ and $(A_{Q_1})^L|(R_{\mathcal{T}_i})^L \cap (X_{A_{Q_1}})^L = (M_{Q_2})^L|(R_{\mathcal{T}_i})^L \cap (X_{M_{Q_2}})^L$, where $i \in \{1, 2\}$. In order to do that we need a few lemmas.

Lemma 5.6. *Let R be a switching of the basic^0 representation \mathcal{T} of a basic cyclic pair, and let $C_{\mathcal{T}} = \emptyset$. Let $R_{\mathcal{T}_1}$ and $R_{\mathcal{T}_2}$ be subsets of R such that $R_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $R_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, $R_{\mathcal{T}_1} \cup R_{\mathcal{T}_2} = R$, $R_{\mathcal{T}_1} \neq \emptyset$, and $R_{\mathcal{T}_2} \neq \emptyset$, where \mathcal{T}_1 and \mathcal{T}_2 are \mathcal{T} 's two maximal subtrees. Let (Q_1, Q_2) be a tree pair of the basic^0 representation \mathcal{T} . Let A_{Q_1} and M_{Q_1} be the two maximal subtrees of Q_1 , and let A_{Q_2} and M_{Q_2} be the two maximal subtrees of Q_2 . If $(R_{\mathcal{T}_j})^L \cap (X_{\Delta_{Q_i}})^L = \emptyset$ then $(R_{\mathcal{T}_m})^L \cap (X_{\Delta_{Q_i}})^L \neq \emptyset$ and $(R_{\mathcal{T}_j})^L \cap (X_{\Diamond_{Q_i}})^L \neq \emptyset$, where $i, j \in \{1, 2\}$, $\Delta \in \{A, M\}$, $m \in \{1, 2\} \setminus \{j\}$, and $\Diamond \in \{A, M\} \setminus \{\Delta\}$.*

Proof. Suppose that $(R_{\mathcal{T}_j})^L \cap (X_{\Delta_{Q_i}})^L = \emptyset$. Since R is a switching of \mathcal{T} , we must have $(X_{\Delta_{Q_i}})^L \subseteq (R_{\mathcal{T}_m})^L$, so that $(R_{\mathcal{T}_m})^L \cap (X_{\Delta_{Q_i}})^L \neq \emptyset$. Since $(R_{\mathcal{T}_j}) \neq \emptyset$, we must have $(R_{\mathcal{T}_j})^L \cap (X_{\Diamond_{Q_i}})^L \neq \emptyset$, as $(R_{\mathcal{T}_j})^L \cap (X_{Q_i})^L \neq \emptyset$, $(R_{\mathcal{T}_j})^L \cap (X_{\Delta_{Q_i}})^L = \emptyset$, and $(X_{Q_i})^L = (X_{A_{Q_i}})^L \cup (X_{M_{Q_i}})^L$. \square

Lemma 5.7. *Let R be a switching of the basic^0 representation \mathcal{T} of a basic cyclic pair, and let $C_{\mathcal{T}} = \emptyset$. Let $R_{\mathcal{T}_1}$ and $R_{\mathcal{T}_2}$ be subsets of R such that $R_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $R_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, $R_{\mathcal{T}_1} \cup R_{\mathcal{T}_2} = R$, $R_{\mathcal{T}_1} \neq \emptyset$, and $R_{\mathcal{T}_2} \neq \emptyset$, where \mathcal{T}_1 and \mathcal{T}_2 are \mathcal{T} 's two maximal subtrees. Let $(\mathcal{T}_1)^L | (R_{\mathcal{T}_1})^L = (\mathcal{T}_2)^L | (R_{\mathcal{T}_1})^L$, and let $(\mathcal{T}_1)^L | (R_{\mathcal{T}_2})^L = (\mathcal{T}_2)^L | (R_{\mathcal{T}_2})^L$. Let (Q_1, Q_2) be a tree pair of the basic^0 representation \mathcal{T} . Let A_{Q_1} and M_{Q_1} be the two maximal subtrees of Q_1 , and let A_{Q_2} and M_{Q_2} be the two maximal subtrees of Q_2 . If $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$, $(X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$, $(X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$, $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$, and the following: $(Q_2)^L | (X_{M_{Q_1}})^L \neq (M_{Q_1})^L$ or $(Q_2)^L | (X_{A_{Q_1}})^L \neq (A_{Q_1})^L$, and $(Q_1)^L | (X_{M_{Q_2}})^L \neq (M_{Q_2})^L$ or $(Q_1)^L | (X_{A_{Q_2}})^L \neq (A_{Q_2})^L$ then either $(R_{\mathcal{T}_1})^L \cap (X_{\Delta})^L \neq \emptyset$ or $(R_{\mathcal{T}_2})^L \cap (X_{\Diamond})^L \neq \emptyset$ for all $\Delta, \Diamond \in (A_{Q_1}, M_{Q_1}, A_{Q_2}, M_{Q_2})$.*

Proof. (Contrapositive.) By Lemma 5.6, we can not have $(R_{\mathcal{T}_1})^L \cap (X_{\Delta})^L = \emptyset$ and $(R_{\mathcal{T}_2})^L \cap (X_{\Delta})^L = \emptyset$, where $\Delta \in (A_{Q_1}, M_{Q_1}, A_{Q_2}, M_{Q_2})$. Suppose we have $(R_{\mathcal{T}_1})^L \cap (X_{\Delta})^L = \emptyset$ and $(R_{\mathcal{T}_2})^L \cap (X_{\Diamond})^L = \emptyset$, where $\Delta \in \{A_{Q_i}, M_{Q_i}\}$, for $i \in \{1, 2\}$, and $\Diamond \in \{A_{Q_i}, M_{Q_i}\} \setminus \{\Delta\}$. Then, by Lemma 5.6, we have $(R_{\mathcal{T}_1})^L \cap (X_{\Diamond})^L \neq \emptyset$ and $(R_{\mathcal{T}_2})^L \cap (X_{\Delta})^L \neq \emptyset$. This implies that $(R_{\mathcal{T}_1})^L \cap (X_{Q_i})^L = (X_{\Diamond})^L$ and $(R_{\mathcal{T}_2})^L \cap (X_{Q_i})^L = (X_{\Delta})^L$. By Lemma 5.5, we have $(Q_1)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = (Q_2)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$ and $(Q_2)^L | (R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L = (Q_1)^L | (R_{\mathcal{T}_2})^L \cap (X_{Q_1})^L$, as $(\mathcal{T}_1)^L | (R_{\mathcal{T}_1})^L = (\mathcal{T}_2)^L | (R_{\mathcal{T}_1})^L$ and $(\mathcal{T}_2)^L | (R_{\mathcal{T}_2})^L = (\mathcal{T}_1)^L | (R_{\mathcal{T}_2})^L$. Since $(X_{Q_1})^L = (X_{Q_2})^L$, we have $(Q_1)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_i})^L = (Q_2)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_i})^L$ and $(Q_2)^L | (R_{\mathcal{T}_2})^L \cap (X_{Q_i})^L = (Q_1)^L | (R_{\mathcal{T}_2})^L \cap (X_{Q_i})^L$, so that $(Q_1)^L | (X_{\Diamond})^L = (Q_2)^L | (X_{\Diamond})^L$ and $(Q_1)^L | (X_{\Delta})^L = (Q_2)^L | (X_{\Delta})^L$; hence either $(Q_2)^L | (X_{M_{Q_1}})^L = (M_{Q_1})^L$ and $(Q_2)^L | (X_{A_{Q_1}})^L = (A_{Q_1})^L$, or $(Q_1)^L | (X_{M_{Q_2}})^L = (M_{Q_2})^L$ and $(Q_1)^L | (X_{A_{Q_2}})^L = (A_{Q_2})^L$.

Finally, suppose that $(R_{\mathcal{T}_1})^L \cap (X_{\Delta_{Q_i}})^L = \emptyset$ and $(R_{\mathcal{T}_2})^L \cap (X_{\Diamond_{Q_j}})^L = \emptyset$, where $\Delta, \Diamond \in \{A, M\}$, $i \in \{1, 2\}$, and $j \in \{1, 2\} \setminus \{i\}$. Take any $x \in (X_{\Delta_{Q_i}})^L$ then $x \notin (R_{\mathcal{T}_1})^L$, but $x \in (R)^L$ and $(R)^L = (R_{\mathcal{T}_1})^L \cup (R_{\mathcal{T}_2})^L$, so $x \in (R_{\mathcal{T}_2})^L$. Since $(R_{\mathcal{T}_2})^L \cap (X_{\Diamond_{Q_j}})^L = \emptyset$, we have $x \notin (X_{\Diamond_{Q_j}})^L$. Therefore

$(X_{\Delta_{Q_i}})^L \cap (X_{\Diamond_{Q_j}})^L = \emptyset$; thus at least one of the following holds: $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L = \emptyset$, $(X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L = \emptyset$, $(X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L = \emptyset$, or $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L = \emptyset$. \square

Proposition 5.4. *Let R be a switching of the basic⁰ representation \mathcal{T} of a basic cyclic pair, and let $C_{\mathcal{T}} = \emptyset$. Let $R_{\mathcal{T}_1}$ and $R_{\mathcal{T}_2}$ be subsets of R such that $R_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $R_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, $R_{\mathcal{T}_1} \cup R_{\mathcal{T}_2} = R$, $R_{\mathcal{T}_1} \neq \emptyset$, and $R_{\mathcal{T}_2} \neq \emptyset$, where \mathcal{T}_1 and \mathcal{T}_2 are \mathcal{T} 's two maximal subtrees. Let $(\mathcal{T}_1)^L | (R_{\mathcal{T}_1})^L = (\mathcal{T}_2)^L | (R_{\mathcal{T}_1})^L$, and let $(\mathcal{T}_1)^L | (R_{\mathcal{T}_2})^L = (\mathcal{T}_2)^L | (R_{\mathcal{T}_2})^L$. Let (Q_1, Q_2) be a tree pair of the basic⁰ representation \mathcal{T} . Let A_{Q_1} and M_{Q_1} be the two maximal subtrees of Q_1 , and let A_{Q_2} and M_{Q_2} be the two maximal subtrees of Q_2 . If $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$, $(X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$, $(X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$, $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$, and the following: $(Q_2)^L | (X_{M_{Q_1}})^L \neq (M_{Q_1})^L$ or $(Q_2)^L | (X_{A_{Q_1}})^L \neq (A_{Q_1})^L$, and $(Q_1)^L | (X_{M_{Q_2}})^L \neq (M_{Q_2})^L$ or $(Q_1)^L | (X_{A_{Q_2}})^L \neq (A_{Q_2})^L$ then either $(R_{\mathcal{T}_i})^L \cap (X_{M_{Q_1}})^L = (R_{\mathcal{T}_i})^L \cap (X_{M_{Q_2}})^L$ and $(R_{\mathcal{T}_i})^L \cap (X_{A_{Q_1}})^L = (R_{\mathcal{T}_i})^L \cap (X_{A_{Q_2}})^L$ or $(R_{\mathcal{T}_i})^L \cap (X_{M_{Q_1}})^L = (R_{\mathcal{T}_i})^L \cap (X_{A_{Q_2}})^L$ and $(R_{\mathcal{T}_i})^L \cap (X_{A_{Q_1}})^L = (R_{\mathcal{T}_i})^L \cap (X_{M_{Q_2}})^L$, for some $i \in \{1, 2\}$.*

Proof. By Lemma 5.5, we have $(Q_1)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = (Q_2)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$ and $(Q_2)^L | (R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L = (Q_1)^L | (R_{\mathcal{T}_2})^L \cap (X_{Q_1})^L$. From Lemma 5.7, either $(R_{\mathcal{T}_1})^L \cap (X_{\Delta})^L \neq \emptyset$ or $(R_{\mathcal{T}_2})^L \cap (X_{\Diamond})^L \neq \emptyset$ for all $\Delta, \Diamond \in (A_{Q_1}, M_{Q_1}, A_{Q_2}, M_{Q_2})$. Without loss of generality, suppose that $(R_{\mathcal{T}_1})^L \cap (X_{\Delta})^L \neq \emptyset$ for all $\Delta \in (A_{Q_1}, M_{Q_1}, A_{Q_2}, M_{Q_2})$. Then the maximal subtrees of $(Q_1)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L$ are $(M_{Q_1})^L | (R_{\mathcal{T}_1})^L \cap (M_{Q_1})^L$ and $(A_{Q_1})^L | (R_{\mathcal{T}_1})^L \cap (A_{Q_1})^L$, as $(R_{\mathcal{T}_1})^L \cap (X_{M_{Q_1}})^L \neq \emptyset$ and $(R_{\mathcal{T}_1})^L \cap (X_{A_{Q_1}})^L \neq \emptyset$. Likewise, the maximal subtrees of $(Q_2)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$ are $(M_{Q_2})^L | (R_{\mathcal{T}_1})^L \cap (M_{Q_2})^L$ and $(A_{Q_2})^L | (R_{\mathcal{T}_1})^L \cap (A_{Q_2})^L$. Since $(Q_1)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = (Q_2)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$, we have either $(M_{Q_1})^L | (R_{\mathcal{T}_1})^L \cap (X_{M_{Q_1}})^L = (M_{Q_2})^L | (R_{\mathcal{T}_1})^L \cap (X_{M_{Q_2}})^L$ and $(A_{Q_1})^L | (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_1}})^L = (A_{Q_2})^L | (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_2}})^L$ or $(M_{Q_1})^L | (R_{\mathcal{T}_1})^L \cap (X_{M_{Q_1}})^L = (A_{Q_2})^L | (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_2}})^L$ and $(A_{Q_1})^L | (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_1}})^L = (M_{Q_2})^L | (R_{\mathcal{T}_1})^L \cap (X_{M_{Q_2}})^L$. Without loss of generality, suppose the former holds then we have $(R_{\mathcal{T}_1})^L \cap (X_{M_{Q_1}})^L = (R_{\mathcal{T}_1})^L \cap (X_{M_{Q_2}})^L$ and $(R_{\mathcal{T}_1})^L \cap (X_{A_{Q_1}})^L = (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_2}})^L$. \square

Now we claim that $(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right)$ or $(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = \left((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$.

Lemma 5.8. *Let R be a switching of the basic⁰ representation \mathcal{T} of a basic cyclic pair, and let $C_{\mathcal{T}} = \emptyset$. Let $R_{\mathcal{T}_1}$ and $R_{\mathcal{T}_2}$ be subsets of R such that $R_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $R_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, and $R_{\mathcal{T}_1} \cup R_{\mathcal{T}_2} = R$, where \mathcal{T}_1 and \mathcal{T}_2 are \mathcal{T} 's two maximal subtrees. Let (Q_1, Q_2) be a tree pair of the basic⁰ representation \mathcal{T} . Let A_{Q_1} and M_{Q_1} be the two maximal subtrees of Q_1 , and let A_{Q_2} and M_{Q_2} be the two maximal*

subtrees of Q_2 . If $(R_{T_i})^L \cap (X_{M_{Q_1}})^L = (R_{T_i})^L \cap (X_{M_{Q_2}})^L$ and $(R_{T_i})^L \cap (X_{A_{Q_1}})^L = (R_{T_i})^L \cap (X_{A_{Q_2}})^L$ or $(R_{T_i})^L \cap (X_{M_{Q_1}})^L = (R_{T_i})^L \cap (X_{A_{Q_2}})^L$ and $(R_{T_i})^L \cap (X_{A_{Q_1}})^L = (R_{T_i})^L \cap (X_{M_{Q_2}})^L$ then $(R_{T_i})^L \cap (X_{Q_1})^L \subseteq \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right)$ or $(R_{T_i})^L \cap (X_{Q_1})^L \subseteq \left((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$, respectively, for $i \in \{1, 2\}$.

Proof. Without loss of generality, suppose that $i = 1$ and that $(R_{T_1})^L \cap (X_{M_{Q_1}})^L = (R_{T_1})^L \cap (X_{M_{Q_2}})^L$ and $(R_{T_1})^L \cap (X_{A_{Q_1}})^L = (R_{T_1})^L \cap (X_{A_{Q_2}})^L$. Take any element x in $(R_{T_1})^L \cap (X_{Q_1})^L$ then $x \in (R_{T_1})^L$ and $x \in (X_{Q_1})^L$. Since $(X_{Q_1})^L = (X_{M_{Q_1}})^L \cup (X_{A_{Q_1}})^L$ and $(X_{M_{Q_1}})^L \cap (X_{A_{Q_1}})^L = \emptyset$, either $x \in (X_{M_{Q_1}})^L$ or $x \in (X_{A_{Q_1}})^L$. Suppose that $x \in (X_{M_{Q_1}})^L$ then $x \in (R_{T_1})^L \cap (X_{M_{Q_1}})^L$. Since $(R_{T_1})^L \cap (X_{M_{Q_1}})^L = (R_{T_1})^L \cap (X_{M_{Q_2}})^L$, we have $x \in (R_{T_1})^L \cap (X_{M_{Q_2}})^L$ thus $x \in (X_{M_{Q_2}})^L$. Hence $x \in (X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L$. If $x \in (X_{A_{Q_1}})^L$ then, similarly, $x \in (X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L$. Therefore $x \in \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right)$. Thus $(R_{T_1})^L \cap (X_{Q_1})^L \subseteq \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right)$. \square

Lemma 5.9. If (Q_1, Q_2) is a tree pair of the basic⁰ representation \mathcal{T} with A_{Q_1} and M_{Q_1} being the two maximal subtrees of Q_1 , and A_{Q_2} , M_{Q_2} being the two maximal subtrees of Q_2 then $(X_{Q_1})^L \setminus \left[\left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \right] = \left((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$.

Proof. Take any $x \in (X_{Q_1})^L \setminus \left[\left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \right]$ then $x \in (X_{Q_1})^L$ and $x \notin \left[\left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \right]$, so $x \notin \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$ and $x \notin \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right)$. Since $(X_{Q_1})^L = (X_{A_{Q_1}})^L \cup (X_{M_{Q_1}})^L$ and $x \in (X_{Q_1})^L$, if $x \in (X_{A_{Q_1}})^L$ then $x \notin (X_{A_{Q_2}})^L$. Since $(X_{Q_1})^L = (X_{Q_2})^L$ and $(X_{Q_2})^L = (X_{A_{Q_2}})^L \cup (X_{M_{Q_2}})^L$, we must have $x \in (X_{M_{Q_2}})^L$, so $x \in \left((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$; hence $x \in \left((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$. Similarly, if $x \in (X_{M_{Q_1}})^L$ then $x \in (X_{A_{Q_2}})^L$, so $x \in \left((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right)$; thus $x \in \left((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$. Therefore $(X_{Q_1})^L \setminus \left[\left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \right] \subseteq \left((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$. Take any $y \in \left((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$ then $y \in \left((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right)$ or $y \in \left((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$. Without loss of generality, suppose that $y \in \left((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right)$ then $y \in (X_{M_{Q_1}})^L$ and $y \in (X_{A_{Q_2}})^L$. Since $(X_{M_{Q_1}})^L \cap (X_{A_{Q_1}})^L = \emptyset$ and $(X_{M_{Q_2}})^L \cap (X_{A_{Q_2}})^L = \emptyset$, $y \notin (X_{A_{Q_1}})^L$ and $y \notin (X_{M_{Q_2}})^L$.

Therefore $y \in (X_{Q_1})^L \setminus \left[\left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \right]$. Therefore we get the result. \square

Proposition 5.5. *Let R be a switching of the basic⁰ representation \mathcal{T} of a basic cyclic pair, and let $C_{\mathcal{T}} = \emptyset$. Let $R_{\mathcal{T}_1}$ and $R_{\mathcal{T}_2}$ be subsets of R such that $R_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $R_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, $R_{\mathcal{T}_1} \cup R_{\mathcal{T}_2} = R$, $R_{\mathcal{T}_1} \neq \emptyset$, and $R_{\mathcal{T}_2} \neq \emptyset$, where \mathcal{T}_1 and \mathcal{T}_2 are \mathcal{T} 's two maximal subtrees. Let $(\mathcal{T}_1)^L | (R_{\mathcal{T}_1})^L = (\mathcal{T}_2)^L | (R_{\mathcal{T}_1})^L$, and let $(\mathcal{T}_1)^L | (R_{\mathcal{T}_2})^L = (\mathcal{T}_2)^L | (R_{\mathcal{T}_2})^L$. Let (Q_1, Q_2) be a tree pair of the basic⁰ representation \mathcal{T} . Let A_{Q_1} and M_{Q_1} be the two maximal subtrees of Q_1 , and let A_{Q_2} and M_{Q_2} be the two maximal subtrees of Q_2 . If $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$, $(X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$, $(X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$, $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$, and the following: $(Q_2)^L | (X_{M_{Q_1}})^L \neq (M_{Q_1})^L$ or $(Q_2)^L | (X_{A_{Q_1}})^L \neq (A_{Q_1})^L$, and $(Q_1)^L | (X_{M_{Q_2}})^L \neq (M_{Q_2})^L$ or $(Q_1)^L | (X_{A_{Q_2}})^L \neq (A_{Q_2})^L$ then $(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right)$ or $(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = \left((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$.*

Proof. By Proposition 5.4 and Lemma 5.8, we get $(R_{\mathcal{T}_i})^L \cap (X_{Q_1})^L \subseteq \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right)$ or $(R_{\mathcal{T}_i})^L \cap (X_{Q_1})^L \subseteq \left((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$, for $i \in \{1, 2\}$. Without loss of generality, suppose that $(R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L \subseteq \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right)$. Take any $x \in \left((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$ then $x \notin (R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L$, by Lemma 5.9, so $x \in (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L$, as R is a switching and $(X_{Q_1})^L = (X_{Q_2})^L$. Therefore $\left((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) \subseteq (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L$. Since $(X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$ and $(X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$, we have $(R_{\mathcal{T}_1})^L \cap (X_{A_{Q_1}})^L \neq \emptyset$, $(R_{\mathcal{T}_1})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$, $(R_{\mathcal{T}_1})^L \cap (X_{M_{Q_1}})^L \neq \emptyset$, and $(R_{\mathcal{T}_1})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$. Thus, by Proposition 5.4 and Lemma 5.8, we have $(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = \left((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) \cup \left((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$. \square

Theorem 5.3. *Let \mathcal{T} be the basic⁰ representation of the basic cyclic pair $N-(w, v, P_1, P_2)$, and let $C_{\mathcal{T}} = \emptyset$. Let \mathcal{T}_1 and \mathcal{T}_2 be \mathcal{T} 's two maximal subtrees. Let R be a switching of the basic⁰ representation \mathcal{T} of the basic cyclic pair, and let W be a switching of the basic⁰ representation \mathcal{T} of the basic cyclic pair. Let $R_{\mathcal{T}_1}$ and $R_{\mathcal{T}_2}$ be subsets of R such that $R_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $R_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, $R_{\mathcal{T}_1} \neq \emptyset$, $R_{\mathcal{T}_2} \neq \emptyset$, and $R_{\mathcal{T}_1} \cup R_{\mathcal{T}_2} = R$, where \mathcal{T}_1 and \mathcal{T}_2 are \mathcal{T} 's two maximal subtrees, and let $W_{\mathcal{T}_1}$ and $W_{\mathcal{T}_2}$ be subsets of W such that $W_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $W_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, and $W_{\mathcal{T}_1} \cup W_{\mathcal{T}_2} = W$. Let (Q_1, Q_2) be any sterile major tree pair of the basic⁰ representation \mathcal{T} . Let A_{Q_1} and M_{Q_1} be the two maximal subtrees of Q_1 , and let A_{Q_2} and M_{Q_2} be the two maximal subtrees of Q_2 . Let $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$, $(X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$, $(X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$, $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$. If $(\mathcal{T}_1 | R_{\mathcal{T}_1})^L = (\mathcal{T}_2 | W_{\mathcal{T}_2})^L$ and $(\mathcal{T}_2 | R_{\mathcal{T}_2})^L =$*

$$\begin{aligned}
(\mathcal{T}_1|W_{\mathcal{T}_1})^L \quad \text{then} \quad (M_{Q_1})^L|((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L) &= (M_{Q_2})^L|((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L), \\
(A_{Q_1})^L|((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L) &= (A_{Q_2})^L|((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L), \\
(M_{Q_2})^L|((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L) &= (A_{Q_1})^L|((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L), \\
(A_{Q_2})^L|((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L) &= (M_{Q_1})^L|((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L).
\end{aligned}$$

Proof. By Lemma 5.4, we have $(\mathcal{T}_1)^L|(R_{\mathcal{T}_1})^L = (\mathcal{T}_2)^L|(R_{\mathcal{T}_1})^L$ and $(\mathcal{T}_2)^L|(R_{\mathcal{T}_2})^L = (\mathcal{T}_1)^L|(R_{\mathcal{T}_2})^L$. Suppose that either $(Q_2)^L|(X_{M_{Q_1}})^L = (M_{Q_1})^L$ and $(Q_2)^L|(X_{A_{Q_1}})^L = (A_{Q_1})^L$, or $(Q_1)^L|(X_{M_{Q_2}})^L = (M_{Q_2})^L$ and $(Q_1)^L|(X_{A_{Q_2}})^L = (A_{Q_2})^L$. Without loss of generality, suppose that $(Q_2)^L|(X_{M_{Q_1}})^L = (M_{Q_1})^L$ and $(Q_2)^L|(X_{A_{Q_1}})^L = (A_{Q_1})^L$. We see that $(A_{Q_1})^L|(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L = ((Q_2)^L|(X_{A_{Q_1}})^L)|(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L$. Now $((Q_2)^L|(X_{A_{Q_1}})^L)|(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L = (Q_2)^L|(X_{A_{Q_1}})^L \cap ((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L)$; hence $(A_{Q_1})^L|(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L = (Q_2)^L|(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L$. It is also the case that $(Q_2)^L|(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L = (A_{Q_2})^L|(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L$. Therefore $(A_{Q_1})^L|(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L = (A_{Q_2})^L|(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L$. In a similar way we get $(M_{Q_1})^L|((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L) = (M_{Q_2})^L|((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L)$, $(M_{Q_2})^L|((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L) = (A_{Q_1})^L|((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L)$, and $(A_{Q_2})^L|((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L) = (M_{Q_1})^L|((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L)$. Suppose that $(Q_2)^L|(X_{M_{Q_1}})^L \neq (M_{Q_1})^L$ or $(Q_2)^L|(X_{A_{Q_1}})^L \neq (A_{Q_1})^L$, and $(Q_1)^L|(X_{M_{Q_2}})^L \neq (M_{Q_2})^L$ or $(Q_1)^L|(X_{A_{Q_2}})^L \neq (A_{Q_2})^L$ then, by Proposition 5.5, we get $(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = ((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L) \cup ((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L)$ or $(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = ((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L) \cup ((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L)$. Without loss of generality, suppose that $(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = ((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L) \cup ((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L)$ then $(R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L = ((X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L) \cup ((X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L)$, as $(R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L = (X_{Q_1})^L \setminus ((R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L)$, $(X_{Q_1})^L = (X_{Q_2})^L$, and Lemma 5.9. By Lemma 5.5, we have $(Q_1)^L|(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = (Q_2)^L|(R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$ and $(Q_2)^L|(R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L = (Q_1)^L|(R_{\mathcal{T}_2})^L \cap (X_{Q_1})^L$. Since $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$, $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$, $(R_{\mathcal{T}_1})^L \cap (X_{A_{Q_1}})^L = ((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L)$, and $(R_{\mathcal{T}_1})^L \cap (X_{M_{Q_1}})^L = ((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L)$, the maximal subtrees of $(Q_1)^L|(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L$ are $(A_{Q_1})^L|(R_{\mathcal{T}_1})^L \cap (X_{A_{Q_1}})^L$ and $(M_{Q_1})^L|(R_{\mathcal{T}_1})^L \cap (X_{M_{Q_1}})^L$. Likewise the maximal subtrees of $(Q_2)^L|(R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$ are $(A_{Q_2})^L|(R_{\mathcal{T}_1})^L \cap (X_{A_{Q_2}})^L$ and $(M_{Q_2})^L|(R_{\mathcal{T}_1})^L \cap (X_{M_{Q_2}})^L$. We must have $(M_{Q_1})^L|(R_{\mathcal{T}_1})^L \cap (X_{M_{Q_1}})^L = (M_{Q_2})^L|(R_{\mathcal{T}_1})^L \cap (X_{M_{Q_2}})^L$ and $(A_{Q_1})^L|(R_{\mathcal{T}_1})^L \cap (X_{A_{Q_1}})^L =$

$(A_{Q_2})^L | (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_2}})^L$, because $(M_{Q_1})^L | (R_{\mathcal{T}_1})^L \cap (X_{M_{Q_1}})^L \neq (A_{Q_2})^L | (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_2}})^L$ and $(A_{Q_1})^L | (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_1}})^L \neq (M_{Q_2})^L | (R_{\mathcal{T}_1})^L \cap (X_{M_{Q_2}})^L$, since $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq (X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L$. Similarly, we must have $(M_{Q_1})^L | (R_{\mathcal{T}_2})^L \cap (X_{M_{Q_1}})^L = (A_{Q_2})^L | (R_{\mathcal{T}_2})^L \cap (X_{A_{Q_2}})^L$ and $(A_{Q_1})^L | (R_{\mathcal{T}_2})^L \cap (X_{A_{Q_1}})^L = (M_{Q_2})^L | (R_{\mathcal{T}_2})^L \cap (X_{M_{Q_2}})^L$. Therefore

$$\begin{aligned}
 (M_{Q_1})^L | \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) &= (M_{Q_2})^L | \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right), \\
 (A_{Q_1})^L | \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) &= (A_{Q_2})^L | \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right), \\
 (M_{Q_2})^L | \left((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \right) &= (A_{Q_1})^L | \left((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \right), \\
 (A_{Q_2})^L | \left((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \right) &= (M_{Q_1})^L | \left((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \right). \quad \square
 \end{aligned}$$

5.3.3.1 Dropping some assumptions

This section explores what happens when we drop the assumption that $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$, $(X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$, $(X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$, and $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$. First, here is a lemma.

Lemma 5.10. *Let \mathcal{T} be the basic^0 representation of the basic cyclic pair $N-(w, v, P_1, P_2)$, and let $C_{\mathcal{T}} = \emptyset$. Let (Q_1, Q_2) be a tree pair of \mathcal{T} . Let A_{Q_1} and M_{Q_1} be the two maximal subtrees of Q_1 , and let A_{Q_2} and M_{Q_2} be the two maximal subtrees of Q_2 . If (Q_1, Q_2) is a sterile major tree pair of \mathcal{T} then we can not have two of the following four sets equal to the empty set: $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L$, $(X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L$, $(X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L$, $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L$.*

Proof. (Contrapositive.) Note that we can not have $(X_{\Delta_{Q_i}})^L \cap (X_{A_{Q_j}})^L = \emptyset$ and $(X_{\Delta_{Q_i}})^L \cap (X_{M_{Q_j}})^L = \emptyset$, where $\Delta \in \{A, M\}$, $i \in \{1, 2\}$, and $j \in \{1, 2\} \setminus \{i\}$, because it implies that $(X_{\Delta_{Q_i}})^L = \emptyset$, which is impossible. Without loss of generality, suppose that $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L = \emptyset$ and $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L = \emptyset$ then $(X_{A_{Q_1}})^L = (X_{M_{Q_2}})^L$ and $(X_{A_{Q_2}})^L = (X_{M_{Q_1}})^L$, which shows that (Q_1, Q_2) is not a sterile major tree pair, by Lemma 5.3. \square

Lemma 5.11. *Let \mathcal{T} be the basic^0 representation of the basic cyclic pair $N-(w, v, P_1, P_2)$, and let $C_{\mathcal{T}} = \emptyset$. Let R be a switching of \mathcal{T} . Let $R_{\mathcal{T}_1}$ and $R_{\mathcal{T}_2}$ be subsets of R such that $R_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $R_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, $R_{\mathcal{T}_1} \neq \emptyset$, $R_{\mathcal{T}_2} \neq \emptyset$, and $R_{\mathcal{T}_1} \cup R_{\mathcal{T}_2} = R$, where \mathcal{T}_1 and \mathcal{T}_2 are \mathcal{T} 's two maximal subtrees. Let (Q_1, Q_2) be a sterile major tree pair of \mathcal{T} . Let A_{Q_1} and M_{Q_1} be the two maximal subtrees of Q_1 , and let A_{Q_2} and M_{Q_2} be the two maximal subtrees of Q_2 . If there exists $x_1, x_2 \in \Lambda$, $y_1, y_2 \in \Upsilon$, and $z_1 \in \Theta$, for $\Lambda \in \Xi$, $\Upsilon \in \Xi \setminus \{\Lambda\}$, and $\Theta \in \Xi \setminus \{\Lambda, \Upsilon\}$, where $\Xi = \left\{ (X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L, (X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L, (X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L, (X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right\}$, such that $x_1 \in (R_{\mathcal{T}_1})^L$, $x_2 \in (R_{\mathcal{T}_2})^L$, $y_1 \in (R_{\mathcal{T}_1})^L$, $y_2 \in (R_{\mathcal{T}_2})^L$, and $z_1 \in (R_{\mathcal{T}_i})^L$, for $i \in \{1, 2\}$, then $(Q_1)^L | (R_{\mathcal{T}_i})^L \cap (X_{Q_1})^L \neq (Q_2)^L | (R_{\mathcal{T}_i})^L \cap (X_{Q_2})^L$.*

Proof. Without loss of generality, suppose that there exists $x_1, x_2 \in \left((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \right)$ such that $x_1 \in (R_{\mathcal{T}_1})^L$ and $x_2 \in (R_{\mathcal{T}_2})^L$, there exists $y_1, y_2 \in \left((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \right)$ such that $y_1 \in (R_{\mathcal{T}_1})^L$ and $y_2 \in (R_{\mathcal{T}_2})^L$, and there exists $z_1 \in \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$ such that $z_1 \in (R_{\mathcal{T}_1})^L$. Now the maximal subtrees of $(Q_1)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L$ are $(A_{Q_1})^L | (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_1}})^L$ and $(M_{Q_1})^L | (R_{\mathcal{T}_1})^L \cap (X_{M_{Q_1}})^L$, and the maximal subtrees of $(Q_2)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$ are $(A_{Q_2})^L | (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_2}})^L$ and $(M_{Q_2})^L | (R_{\mathcal{T}_1})^L \cap (X_{M_{Q_2}})^L$. We see that $(A_{Q_1})^L | (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_1}})^L \neq (A_{Q_2})^L | (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_2}})^L$, because $x_1 \in (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_1}})^L$ and $x_1 \in (X_{M_{Q_2}})^L$, so that $x_1 \notin (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_2}})^L$ as $(X_{M_{Q_2}})^L \cap (X_{A_{Q_2}})^L = \emptyset$; thus $(R_{\mathcal{T}_1})^L \cap (X_{A_{Q_1}})^L \neq (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_2}})^L$. We also see that $(A_{Q_1})^L | (R_{\mathcal{T}_1})^L \cap (X_{A_{Q_1}})^L \neq (M_{Q_2})^L | (R_{\mathcal{T}_1})^L \cap (X_{M_{Q_2}})^L$, because $z_1 \in (R_{\mathcal{T}_1})^L \cap (X_{M_{Q_2}})^L$ and $z_1 \in (X_{M_{Q_1}})^L$, so that $z_1 \notin (X_{A_{Q_1}})^L$ as $(X_{A_{Q_1}})^L \cap (X_{M_{Q_1}})^L = \emptyset$; thus $(R_{\mathcal{T}_1})^L \cap (X_{A_{Q_1}})^L \neq (R_{\mathcal{T}_1})^L \cap (X_{M_{Q_2}})^L$. Therefore $(Q_1)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L \neq (Q_2)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$. \square

Theorem 5.4. *Let \mathcal{T} be the basic⁰ representation of the basic cyclic pair $N-(w, v, P_1, P_2)$, and let $C_{\mathcal{T}} = \emptyset$. Let \mathcal{T}_1 and \mathcal{T}_2 be the two maximal subtrees of \mathcal{T} such that $(\mathcal{T}_1)^L \neq (\mathcal{T}_2)^L$. Let (Q_1, Q_2) be a sterile major tree pair of \mathcal{T} . Let A_{Q_1} and M_{Q_1} be the two maximal subtrees of Q_1 , and let A_{Q_2} and M_{Q_2} be the two maximal subtrees of Q_2 . Let $(\mathcal{T}_2)^L | (X_{M_{\mathcal{T}_1}})^L \neq (M_{\mathcal{T}_1})^L$ and $(\mathcal{T}_1)^L | (X_{M_{\mathcal{T}_2}})^L \neq (M_{\mathcal{T}_2})^L$. If two of the following four equations hold then \mathcal{T} does not display a tree twice:*

$$\begin{aligned} (M_{Q_1})^L | \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) &\neq (M_{Q_2})^L | \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right), \\ (A_{Q_1})^L | \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) &\neq (A_{Q_2})^L | \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right), \\ (M_{Q_2})^L | \left((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \right) &\neq (A_{Q_1})^L | \left((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \right), \\ (A_{Q_2})^L | \left((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \right) &\neq (M_{Q_1})^L | \left((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \right). \end{aligned}$$

Proof. Without loss of generality, suppose that $(M_{Q_2})^L | \left((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \right) \neq (A_{Q_1})^L | \left((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \right)$ and $(A_{Q_2})^L | \left((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \right) \neq (M_{Q_1})^L | \left((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \right)$. Then $(X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \neq \emptyset$ and $(X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \neq \emptyset$. Since (Q_1, Q_2) is a sterile major tree pair and Lemma 5.10, $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \neq \emptyset$ or $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$. Without loss of generality, suppose that $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$. Suppose for contradiction that \mathcal{T} displays a tree twice then there exists two switchings R and W , of \mathcal{T} , such that $(\mathcal{T}|R)^L = (\mathcal{T}|W)^L$. Let $R_{\mathcal{T}_1}$ and $R_{\mathcal{T}_2}$ be subsets of R such that $R_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $R_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, and $R_{\mathcal{T}_1} \cup R_{\mathcal{T}_2} = R$, and let $W_{\mathcal{T}_1}$ and $W_{\mathcal{T}_2}$ be subsets of W such that $W_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $W_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, and $W_{\mathcal{T}_1} \cup W_{\mathcal{T}_2} = W$. We must have $R_{\mathcal{T}_1}$, $R_{\mathcal{T}_2}$, $W_{\mathcal{T}_1}$, and $W_{\mathcal{T}_2}$ all non-empty, because

$(\mathcal{T}_1)^L \neq (\mathcal{T}_2)^L$, $(\mathcal{T}_2)^L | (X_{M_{\mathcal{T}_1}})^L \neq (M_{\mathcal{T}_1})^L$, and $(\mathcal{T}_1)^L | (X_{M_{\mathcal{T}_2}})^L \neq (M_{\mathcal{T}_2})^L$. By Lemma 5.4, $(\mathcal{T}_1)^L | (R_{\mathcal{T}_1})^L = (\mathcal{T}_2)^L | (R_{\mathcal{T}_1})^L$ and $(\mathcal{T}_2)^L | (R_{\mathcal{T}_2})^L = (\mathcal{T}_1)^L | (R_{\mathcal{T}_2})^L$, because $(\mathcal{T}_1 | R_{\mathcal{T}_1})^L = (\mathcal{T}_2 | W_{\mathcal{T}_2})^L$ and $(\mathcal{T}_2 | R_{\mathcal{T}_2})^L = (\mathcal{T}_1 | W_{\mathcal{T}_1})^L$ is implied by $(\mathcal{T} | R)^L = (\mathcal{T} | W)^L$ and $R_{\mathcal{T}_1}$, $R_{\mathcal{T}_2}$, $W_{\mathcal{T}_1}$, and $W_{\mathcal{T}_2}$ all being non-empty. If $(X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \subseteq (R_{\mathcal{T}_i})^L$ or $(X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \subseteq (R_{\mathcal{T}_i})^L$, for $i \in \{1, 2\}$, then we reach a contradiction, which we now show. Without loss of generality, suppose that $(X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \subseteq (R_{\mathcal{T}_1})^L$. By Lemma 5.5, $(Q_1)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = (Q_2)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$, and this implies that $(Q_1)^L | (X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L = (Q_2)^L | (X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L$, as $(X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \subseteq (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L$ and $(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$. Now that implies that $(M_{Q_2})^L | ((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L) = (A_{Q_1})^L | ((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L)$ - a contradiction. Therefore we can not have $(X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \subseteq (R_{\mathcal{T}_i})^L$ or $(X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \subseteq (R_{\mathcal{T}_i})^L$, for $i \in \{1, 2\}$. Thus $(X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L$ and $(X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L$ are split between $(R_{\mathcal{T}_1})^L$ and $(R_{\mathcal{T}_2})^L$, so there exists $x_1, x_2 \in ((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L)$ such that $x_1 \in (R_{\mathcal{T}_1})^L$ and $x_2 \in (R_{\mathcal{T}_2})^L$, and there exists $y_1, y_2 \in ((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L)$ such that $y_1 \in (R_{\mathcal{T}_1})^L$ and $y_2 \in (R_{\mathcal{T}_2})^L$. Now $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \neq \emptyset$, so either $(R_{\mathcal{T}_1})^L \cap ((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L) \neq \emptyset$ or $(R_{\mathcal{T}_2})^L \cap ((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L) \neq \emptyset$. Without loss of generality, suppose that $(R_{\mathcal{T}_1})^L \cap ((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L) \neq \emptyset$ then there exists $z_1 \in ((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L)$ such that $z_1 \in (R_{\mathcal{T}_1})^L$. By Lemma 5.11, $(Q_1)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L \neq (Q_2)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$ - a contradiction. Hence \mathcal{T} does not display a tree twice. \square

5.3.3.2 An iterative algorithm

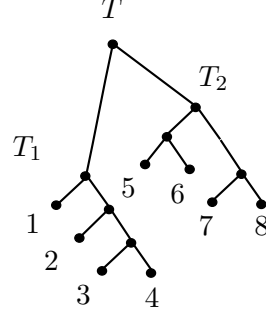
We are now ready to tackle the case where exactly one of $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L$, $(X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L$, $(X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L$, $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L$ is the empty set and exactly one of $(M_{Q_1})^L | ((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L) \neq (M_{Q_2})^L | ((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L)$, $(A_{Q_1})^L | ((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L) \neq (A_{Q_2})^L | ((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L)$, $(M_{Q_2})^L | ((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L) \neq (A_{Q_1})^L | ((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L)$, $(A_{Q_2})^L | ((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L) \neq (M_{Q_1})^L | ((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L)$ holds. We characterise this case in terms of an iterative algorithm. The algorithm uses a new concept called “structural soundness.”

Definition 5.9. Let T be a tree on leaf set X . We say that a leaf set X_1 is *structurally sound* in T when there exists a subtree T_1 of T such that X_1 is the leaf set of T_1 .

Note 7. If X_1 is structurally sound in a tree T then the smallest subtree in T containing all of X_1

contains nothing but X_1 .

Example 5.8. The leaf set $\{5, 7, 8\}$ is not structurally sound in T . The leaf set $\{2, 3, 4\}$ is structurally sound in T .



Definition 5.10. Let T be a tree on leaf set X . Let T_1 be a subtree of T such that $T_1 \neq T$. Let the root of T_1 be x_1 , and let the parent of x_1 be u . Let the other child of u be y . Let T_2 be the subtree of T whose root is y . The subtree T_2 is called the *sibling* of T_1 .

Algorithm : ITERATIVEALG

Input : A basic⁰ representation \mathcal{T} representing the basic cyclic pair $B-(w, v, P_1, P_2)$ such that $C_{\mathcal{T}} = \emptyset$. A sterile major tree pair (Q_1, Q_2) of \mathcal{T} , with A_{Q_1}, M_{Q_1} being the two maximal subtrees of Q_1 , and A_{Q_2}, M_{Q_2} being the two maximal subtrees of Q_2 , such that exactly one of $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L$, $(X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L$, $(X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L$, $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L$ is the empty set, call it $(X_B)^L$, and exactly one of $(M_{Q_1})^L \mid ((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L) \neq (M_{Q_2})^L \mid ((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L)$, $(A_{Q_1})^L \mid ((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L) \neq (A_{Q_2})^L \mid ((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L)$, $(M_{Q_2})^L \mid ((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L) \neq (A_{Q_1})^L \mid ((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L)$, $(A_{Q_2})^L \mid ((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L) \neq (M_{Q_1})^L \mid ((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L)$ holds, say $(\Delta_{Q_1})^L \mid (X_{Z_1})^L \neq (\Diamond_{Q_2})^L \mid (X_{Z_1})^L$, where $\Delta, \Diamond \in \{A, M\}$ and $(X_{Z_1})^L = (X_{\Delta_{Q_1}})^L \cap (X_{\Diamond_{Q_2}})^L$. We also have two sets $(R_{Q_1}^{(1)*})^L \in \Xi \setminus \{(X_B)^L, (X_{Z_1})^L\}$ and $(R_{Q_2}^{(1)*})^L \in \Xi \setminus \{(X_B)^L, (X_{Z_1})^L, (R_{Q_1}^{(1)*})^L\}$, where $\Xi = \{(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L, (X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L, (X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L, (X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L\}$.

We have indices $i, j \in \mathbb{N}$, and we have a function $f(n) = \begin{cases} 2 & \text{when } n \text{ is odd} \\ 1 & \text{when } n \text{ is even} \end{cases}$, where $n \in \mathbb{N}$.

Output : The statement “displays a tree twice,” the statement “does not display a tree twice,” or

a major tree pair $(U_{Q_{f(j+1)}}, U_{Q_{f(j)}})$ of \mathcal{T} together with sets $(R_{Q_1}^{(j)*})^L$, $(R_{Q_2}^{(j)*})^L$, and $(X_{Z_j})^L$, for some $j \in \mathbb{N}$.

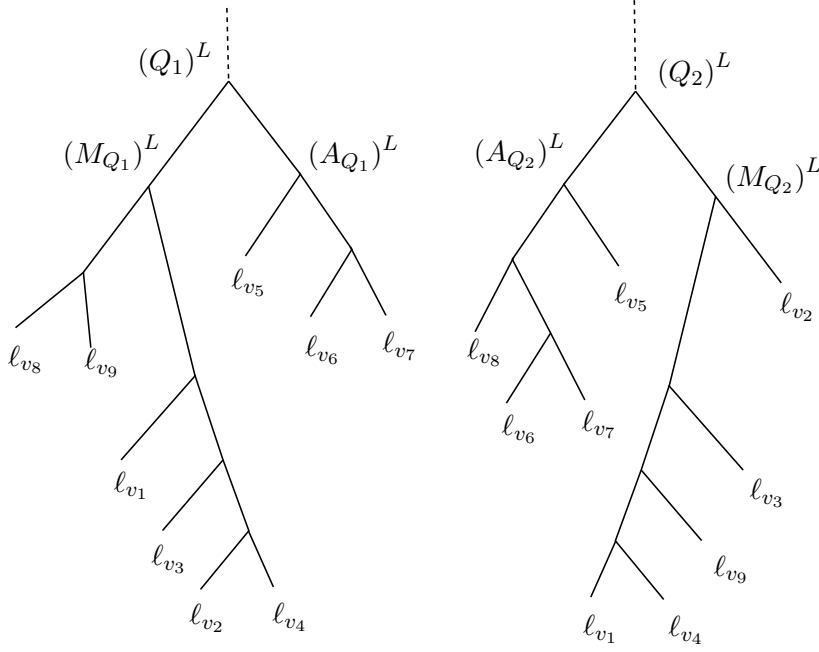
1. Choose $(R_{Q_1}^{(1)*})^L$ and $(R_{Q_2}^{(1)*})^L$ such that $(X_{Z_1})^L$ is structurally sound in $(Q_1)^L \mid (R_{Q_2}^{(1)*})^L \cup (X_{Z_1})^L$ and $(Q_2)^L \mid (R_{Q_1}^{(1)*})^L \cup (X_{Z_1})^L$.
2. In $(Q_2)^L \mid (R_{Q_2}^{(1)*})^L \cup (X_{Z_1})^L$ check if $(X_{Z_1})^L$ is structurally sound - if so then go to Step 3 with $j = 2$ and $i = 1$, and set $(R_{Q_1}^{(2)*})^L = (R_{Q_1}^{(1)*})^L$, $(R_{Q_2}^{(2)*})^L = (R_{Q_2}^{(1)*})^L$, and $(X_{Z_2})^L = (X_{Z_1})^L$. Otherwise go to Step 3 with $j = 1$ and $i = 1$.
3. In $(Q_{f(j+1)})^L \mid (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$ find the subtree whose leaf set is $(X_{Z_j})^L$, call it Z_j . We then find the sibling of Z_j , call it $(S_{Z_j})^L$.
4. In $(Q_{f(j)})^L \mid (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$ we find the smallest subtree containing all of $(X_{S_{Z_j}})^L$, where $(X_{S_{Z_j}})^L$ is the leaf set of $(S_{Z_j})^L$, call it D_{i-1} .
5. Let $(R_{Q_{f(j+1)}}^i)^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup ((X_{D_{i-1}}) \setminus (X_{S_{Z_j}})^L)$, where $(X_{D_{i-1}})$ is the leaf set of D_{i-1} . If $(Q_1)^L \mid (R_{Q_{f(j+1)}}^i)^L \neq (Q_2)^L \mid (R_{Q_{f(j+1)}}^i)^L$ then go to Step 12.
6. Look at $(Q_{f(j)})^L \mid (R_{Q_{f(j)}}^i)^L$, where $(R_{Q_{f(j)}}^i)^L = (X_{Q_{f(j)}})^L \setminus (R_{Q_{f(j+1)}}^i)^L$. If $(Q_1)^L \mid (R_{Q_{f(j)}}^i)^L = (Q_2)^L \mid (R_{Q_{f(j)}}^i)^L$ then go to Step 11. In $(Q_{f(j)})^L \mid (R_{Q_{f(j)}}^i)^L$ the smallest subtree containing all of $(X_{S_{Z_j}})^L$ contains nothing but $(X_{S_{Z_j}})^L$, call it D_i .
7. Find the sibling of D_i in $(Q_{f(j)})^L \mid (R_{Q_{f(j)}}^i)^L$, call it S_{D_i} . Let $(X_{S_{D_i}})^L$ be the leaf set of S_{D_i} . If $(X_{S_{D_i}})^L \not\subseteq (X_{Z_j})^L$ then go to Step 12.
8. If $(X_{S_{D_i}})^L = (X_{Z_j})^L$ then $(U_{Q_{f(j+1)}}, U_{Q_{f(j)}})$, where $U_{Q_{f(j+1)}}$ is the subtree of $Q_{f(j+1)}$ such that $(X_{U_{Q_{f(j+1)}}})^L = (X_{Z_j})^L$ and $U_{Q_{f(j)}}$ is the subtree of $Q_{f(j)}$ such that $(X_{U_{Q_{f(j)}}})^L = (X_{Z_j})^L$, is a major tree pair of \mathcal{T} , so return the major tree pair $(U_{Q_{f(j+1)}}, U_{Q_{f(j)}})$ together with the sets $(R_{Q_1}^{(j)*})^L$, $(R_{Q_2}^{(j)*})^L$, and $(X_{Z_j})^L$.
9. Check if $(Q_1)^L \mid (R_{Q_{f(j)}}^{i+1})^L = (Q_2)^L \mid (R_{Q_{f(j)}}^{i+1})^L$, where $(R_{Q_{f(j)}}^{i+1})^L = ((R_{Q_{f(j)}}^i)^L \setminus (X_{Z_j})^L) \cup (X_{S_{D_i}})^L$. If not then go to Step 10. Check if $(Q_1)^L \mid (R_{Q_{f(j+1)}}^{i+1})^L = (Q_2)^L \mid (R_{Q_{f(j+1)}}^{i+1})^L$, where $(R_{Q_{f(j+1)}}^{i+1})^L = (X_{Q_{f(j+1)}})^L \setminus (R_{Q_{f(j)}}^{i+1})^L$. If it does then go to Step 11. Check if $(Q_1)^L \mid (R_{Q_{f(j+1)}}^{i+2})^L = (Q_2)^L \mid (R_{Q_{f(j+1)}}^{i+2})^L$, where $(R_{Q_{f(j+1)}}^{i+2})^L = (R_{Q_{f(j+1)}}^{i+1})^L \cup (X_{S_{D_i}})^L$. If not then go to Step 12, but if it does then go back to Step 6 replacing i with $i + 2$.

10. We have $(Q_1)^L \mid (R_{Q_{f(j)}}^m)^L \neq (Q_2)^L \mid (R_{Q_{f(j)}}^m)^L$, where $(R_{Q_{f(j)}}^m)^L = \left((R_{Q_{f(j)}}^{m-1})^L \setminus (X_{Z_j})^L \right) \cup (X_{S_{D_{m-1}}})^L$, for some $m \in \mathbb{N}$ such that $m \geq i + 1$. Check if $(Q_1)^L \mid (R_{Q_{f(j+1)}}^m)^L = (Q_2)^L \mid (R_{Q_{f(j+1)}}^m)^L$, where $(R_{Q_{f(j+1)}}^m)^L = (X_{Q_{f(j+1)}})^L \setminus (R_{Q_{f(j)}}^m)^L$. If not then go to Step 12. If so then go to Step 3 with j replaced by $j + 1$, and set $(R_{Q_{f(j+1)}}^{(j+1)*})^L = (R_{Q_{f(j+1)}}^m)^L$, $(R_{Q_{f(j+2)}}^{(j+1)*})^L = (R_{Q_{f(j)}}^{(j)*})^L$, and $(X_{Z_{j+1}})^L = (X_{S_{D_{m-1}}})^L$. We also set $i = m + 1$.
11. Return the statement “displays a tree twice.”
12. Return the statement “does not display a tree twice.”

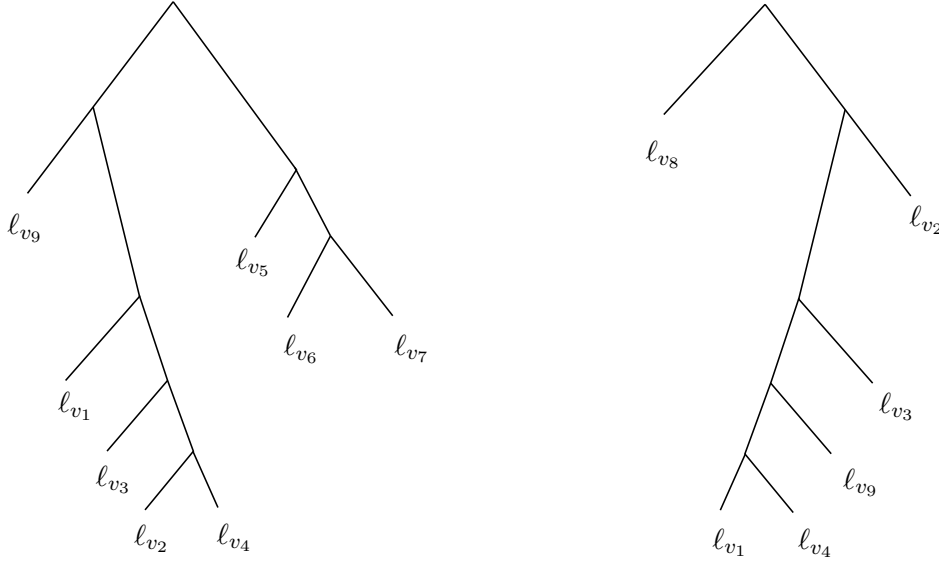
5.3.3.3 How the iterative algorithm works

The idea behind ITERATIVEALG is to preserve the equations $(Q_1)^L \mid (R_{Q_{f(j)}}^{(j)*})^L = (Q_2)^L \mid (R_{Q_{f(j)}}^{(j)*})^L$ and $(Q_1)^L \mid (R_{Q_{f(j+1)}}^{(j)*})^L = (Q_2)^L \mid (R_{Q_{f(j+1)}}^{(j)*})^L$, whilst at the same time seeing what from $(X_{Z_j})^L$ can go with $(R_{Q_{f(j)}}^{(j)*})^L$ and what can go with $(R_{Q_{f(j+1)}}^{(j)*})^L$. If we find some part of $(X_{Z_j})^L$ that can not go with $(R_{Q_{f(j)}}^{(j)*})^L$, i.e. when part of $(X_{Z_j})^L$ causes loss of equality due to lack of structural soundness, then we see if that troublesome part of $(X_{Z_j})^L$ can go with $(R_{Q_{f(j+1)}}^{(j)*})^L$. If that does not work then we know that we will never display a tree twice, because the troublesome part has to go somewhere and we can not split the troublesome part up and split up the other part of $(X_{Z_j})^L$, by Lemma 5.11. On the other hand if the troublesome part can go with $(R_{Q_{f(j+1)}}^{(j)*})^L$ then we have expanded $(R_{Q_{f(j+1)}}^{(j)*})^L$ whilst preserving equality, so we check if $(R_{Q_{f(j)}}^{(j)*})^L$ together with $(X_{Z_j})^L$ minus the troublesome part preserves equality. If it does then we know that we can choose $(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L$ and $(R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L$ such that $(Q_1)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = (Q_2)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$ and $(Q_1)^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_1})^L = (Q_2)^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L$, so we are on our way to displaying a tree twice. If $(R_{Q_{f(j)}}^{(j)*})^L$ together with $(X_{Z_j})^L$ minus the troublesome part does not work then we need to split up $(X_{Z_j})^L$ minus the troublesome part. In this way we make some progress, and the algorithm must eventually stop, because $(X_{Z_j})^L$ is finite.

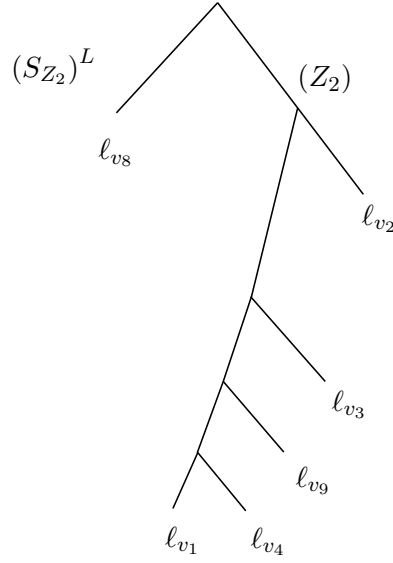
Example 5.9. Here is a worked example showing how ITERATIVEALG works. We see that $(X_B)^L = (X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L = \emptyset$ and $(X_{Z_1})^L = (X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L = \{\ell_{v_1}, \ell_{v_2}, \ell_{v_3}, \ell_{v_4}, \ell_{v_9}\}$. We choose $(R_{Q_1}^{(1)*})^L = (X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L = \{\ell_{v_8}\}$ and $(R_{Q_2}^{(1)*})^L = (X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L = \{\ell_{v_5}, \ell_{v_6}, \ell_{v_7}\}$, because $(X_{Z_1})^L$ is structurally sound in $(Q_1)^L \mid (R_{Q_2}^{(1)*})^L \cup (X_{Z_1})^L$ and $(Q_2)^L \mid (R_{Q_1}^{(1)*})^L \cup (X_{Z_1})^L$.



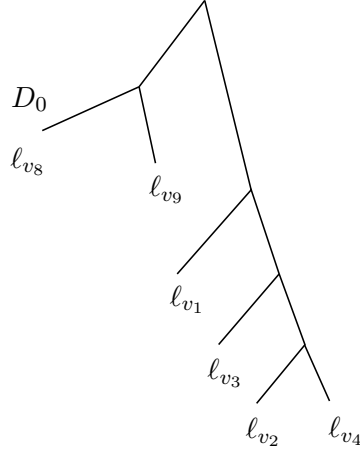
Here we see $(Q_1)^L \mid (R_{Q_2}^{(1)*})^L \cup (X_{Z_1})^L$ shown on the left and $(Q_2)^L \mid (R_{Q_1}^{(1)*})^L \cup (X_{Z_1})^L$ shown on the right.



In Step 2, we see that $(X_{Z_1})^L$ is structurally sound in $(Q_2)^L \mid (R_{Q_2}^{(1)*})^L \cup (X_{Z_1})^L$, so we set $(R_{Q_1}^{(2)*})^L = (R_{Q_1}^{(1)*})^L$, $(R_{Q_2}^{(2)*})^L = (R_{Q_2}^{(1)*})^L$, and $(X_{Z_2})^L = (X_{Z_1})^L$, and we go to Step 3 with $j = 2$ and $i = 1$. Now we see that $f(j+1) = 2$ and $f(j) = 1$. Hence in Step 3 we are looking at $(Q_2)^L \mid (R_{Q_1}^{(2)*})^L \cup (X_{Z_2})^L$, and the sibling of $(X_{Z_2})^L$ is the leaf ℓ_{v_8} . Shown is the result of Step 3.



In Step 4 we look at $(Q_1)^L \mid (R_{Q_1}^{(2)*})^L \cup (X_{Z_2})^L$, and shown is the result of Step 4.



After two iterations, we find that $(R_{Q_2}^6)^L = \{\ell_{v_5}, \ell_{v_6}, \ell_{v_7}, \ell_{v_9}, \ell_{v_1}, \ell_{v_4}\}$ works and $(R_{Q_1}^6)^L = \{\ell_{v_2}, \ell_{v_3}, \ell_{v_8}\}$ works, so that the algorithm outputs the statement “displays a tree twice.”

Lemma 5.12. *Let \mathcal{T} be the basic⁰ representation of the basic cyclic pair $N-(w, v, P_1, P_2)$, and let $C_{\mathcal{T}} = \emptyset$. Let (Q_1, Q_2) be a sterile major tree pair with A_{Q_1}, M_{Q_1} being the two maximal subtrees of Q_1 , and A_{Q_2}, M_{Q_2} being the two maximal subtrees of Q_2 . If exactly one of $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L$, $(X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L$, $(X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L$, $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L$ is the empty set, call it $(X_B)^L$, and exactly one of $(M_{Q_1})^L \mid ((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L) \neq (M_{Q_2})^L \mid ((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L)$, $(A_{Q_1})^L \mid ((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L) \neq (A_{Q_2})^L \mid ((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L)$, $(M_{Q_2})^L \mid ((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L) \neq (A_{Q_1})^L \mid ((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L)$, $(A_{Q_2})^L \mid ((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L) \neq (M_{Q_1})^L \mid ((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L)$ holds, say $(\triangle_{Q_1})^L \mid (X_{Z_1})^L \neq (\diamond_{Q_2})^L \mid (X_{Z_1})^L$, where $\triangle, \diamond \in \{A, M\}$ and $(X_{Z_1})^L = (X_{\triangle_{Q_1}})^L \cap (X_{\diamond_{Q_2}})^L$,*

then $(X_{Z_1})^L$ is structurally sound in both $(Q_1)^L \mid (R_{Q_2}^{(1)*})^L \cup (X_{Z_1})^L$ and $(Q_2)^L \mid (R_{Q_1}^{(1)*})^L \cup (X_{Z_1})^L$ where $(R_{Q_1}^{(1)*})^L \in \Xi \setminus \{(X_B)^L, (X_{Z_1})^L\}$ and $(R_{Q_2}^{(1)*})^L \in \Xi \setminus \{(X_B)^L, (X_{Z_1})^L, (R_{Q_1}^{(1)*})^L\}$, and $\Xi = \{(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L, (X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L, (X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L, (X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L\}$.

Proof. Without loss of generality, suppose that $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L = \emptyset$ and $(M_{Q_1})^L \mid ((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L) \neq (M_{Q_2})^L \mid ((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L)$ then $(X_{A_{Q_1}})^L \subseteq (X_{M_{Q_2}})^L$ and $(X_{A_{Q_2}})^L \subseteq (X_{M_{Q_1}})^L$, so $(X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L = (X_{A_{Q_1}})^L$ and $(X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L = (X_{A_{Q_2}})^L$; hence we want to show that $(X_{Z_1})^L = (X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L$ is structurally sound in either $(Q_1)^L \mid (X_{A_{Q_1}})^L \cup (X_{Z_1})^L$ and $(Q_2)^L \mid (X_{A_{Q_2}})^L \cup (X_{Z_1})^L$ or $(Q_1)^L \mid (X_{A_{Q_2}})^L \cup (X_{Z_1})^L$ and $(Q_2)^L \mid (X_{A_{Q_1}})^L \cup (X_{Z_1})^L$. It is fairly clear that $(X_{Z_1})^L$ is structurally sound in $(Q_1)^L \mid (X_{A_{Q_1}})^L \cup (X_{Z_1})^L$ and $(Q_2)^L \mid (X_{A_{Q_2}})^L \cup (X_{Z_1})^L$, as the maximal subtrees of $(Q_1)^L \mid (X_{A_{Q_1}})^L \cup (X_{Z_1})^L$ are $(A_{Q_1})^L$ and $(M_{Q_1})^L \mid (X_{Z_1})^L$ because $(X_{A_{Q_1}})^L \cap (X_{Z_1})^L = \emptyset$. Likewise in $(Q_2)^L \mid (X_{A_{Q_2}})^L \cup (X_{Z_1})^L$, so $(R_{Q_1}^{(1)*})^L$ is chosen to be $(X_{A_{Q_2}})^L$ and $(R_{Q_2}^{(1)*})^L$ is chosen to be $(X_{A_{Q_1}})^L$. The other cases are similar. \square

Lemma 5.13. In ITERATIVEALG, if $(X_{Q_1})^L = (X_{Q_2})^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$, $(R_{Q_{f(j+1)}}^{(j)*})^L \cap (R_{Q_{f(j)}}^{(j)*})^L = \emptyset$, $(R_{Q_{f(j+1)}}^{(j)*})^L \cap (X_{Z_j})^L = \emptyset$, and $(R_{Q_{f(j)}}^{(j)*})^L \cap (X_{Z_j})^L = \emptyset$, for all $j \in \mathbb{N}$, then $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{Q_{f(j+1)}}^i)^L$ and $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{Q_{f(j)}}^i)^L$, for all $i, j \in \mathbb{N}$ such that $j \leq i$.

Proof. In Step 5 of the algorithm, $(R_{Q_{f(j+1)}}^i)^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup ((X_{D_{i-1}}) \setminus (X_{S_{Z_j}}))^L$, where $i, j \in \mathbb{N}$, so that $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{Q_{f(j+1)}}^i)^L$. In Step 6, $(R_{Q_{f(j)}}^i)^L = (X_{Q_{f(j)}})^L \setminus (R_{Q_{f(j+1)}}^i)^L$, so $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{Q_{f(j)}}^i)^L$, because $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{Q_{f(j+1)}}^i)^L$ and $(X_{Q_{f(j)}})^L \setminus (R_{Q_{f(j+1)}}^i)^L = (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$ (the latter follows from the hypothesis). Therefore $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{Q_{f(j)}}^i)^L$ and $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{Q_{f(j+1)}}^i)^L$. Now in Step 9 we see that $(R_{Q_{f(j)}}^{i+1})^L = ((R_{Q_{f(j)}}^i)^L \setminus (X_{Z_j})^L) \cup (X_{S_{D_i}})^L$. Since $(X_{Z_j})^L \cap (R_{Q_{f(j)}}^{(j)*})^L = \emptyset$ and $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{Q_{f(j)}}^i)^L$, we have $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{Q_{f(j)}}^{i+1})^L$. Also $(R_{Q_{f(j+1)}}^{i+1})^L = (X_{Q_{f(j+1)}})^L \setminus (R_{Q_{f(j)}}^{i+1})^L$, so that $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{Q_{f(j+1)}}^{i+1})^L$. It follows that $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{Q_{f(j+1)}}^i)^L$ and $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{Q_{f(j)}}^i)^L$, for all $i, j \in \mathbb{N}$ such that $j \leq i$. \square

Lemma 5.14. In ITERATIVEALG, $(X_{Q_1})^L = (X_{Q_2})^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$, $(R_{Q_{f(j+1)}}^{(j)*})^L \cap (R_{Q_{f(j)}}^{(j)*})^L = \emptyset$, $(R_{Q_{f(j+1)}}^{(j)*})^L \cap (X_{Z_j})^L = \emptyset$, and $(R_{Q_{f(j)}}^{(j)*})^L \cap (X_{Z_j})^L = \emptyset$, for all $j \in \mathbb{N}$. Moreover, $(R_{Q_{f(j+1)}}^{(j+1)*})^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup ((X_{Z_j})^L \setminus (X_{Z_{j+1}})^L)$.

Proof. We will prove the lemma by using induction on j . Base case $j = 1$: When $j = 1$ we have $(R_{Q_1}^{(1)*})^L$, $(R_{Q_2}^{(1)*})^L$, and $(X_{Z_1})^L$. By definition, $(R_{Q_1}^{(1)*})^L \in$

$\Xi \setminus \{(X_B)^L, (X_{Z_1})^L\}$ and $(R_{Q_2}^{(1)*})^L \in \Xi \setminus \{(X_B)^L, (X_{Z_1})^L, (R_{Q_1}^{(1)*})^L\}$, where $\Xi = \{(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L, (X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L, (X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L, (X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L\}$.

Also $(X_{Z_1})^L \in \Xi$. It is straight forward to show that $\Theta \cap \Lambda = \emptyset$ where $\Theta, \Lambda \in \Xi$. Therefore $(R_{Q_1}^{(1)*})^L \cap (R_{Q_2}^{(1)*})^L = \emptyset$, $(R_{Q_2}^{(1)*})^L \cap (X_{Z_1})^L = \emptyset$, and $(R_{Q_1}^{(1)*})^L \cap (X_{Z_1})^L = \emptyset$. We know that $(X_{Q_1})^L = (X_{Q_2})^L$, as (Q_1, Q_2) is a sterile major tree pair, and it is easy to show that $(X_{Q_1})^L = \bigcup \Xi$, using Lemma 5.9. Hence $(X_{Q_1})^L = (X_{Q_2})^L = (R_{Q_1}^{(1)*})^L \cup (R_{Q_2}^{(1)*})^L \cup (X_{Z_1})^L$, as $(X_B)^L = \emptyset$. Induction step: Suppose that $(X_{Q_1})^L = (X_{Q_2})^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$, $(R_{Q_{f(j+1)}}^{(j)*})^L \cap (R_{Q_{f(j)}}^{(j)*})^L = \emptyset$, $(R_{Q_{f(j+1)}}^{(j)*})^L \cap (X_{Z_j})^L = \emptyset$, and $(R_{Q_{f(j)}}^{(j)*})^L \cap (X_{Z_j})^L = \emptyset$ then, by Lemma 5.13, $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{Q_{f(j+1)}}^i)^L$ and $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{Q_{f(j)}}^i)^L$, for all $j \in \mathbb{N}$, and for all $i \in \mathbb{N}$. The only way we can get $(R_{Q_{f(j+1)}}^{(j+1)*})^L$, $(R_{Q_{f(j+2)}}^{(j+1)*})^L$, and $(X_{Z_{j+1}})^L$ is when we get to Step 10. In Step 10, $(R_{Q_{f(j+1)}}^{(j+1)*})^L = (R_{Q_{f(j+1)}}^m)^L$, $(R_{Q_{f(j+2)}}^{(j+1)*})^L = (R_{Q_{f(j)}}^{(j)*})^L$, and $(X_{Z_{j+1}})^L = (X_{S_{D_{m-1}}})^L$. Since $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{Q_{f(j+1)}}^i)^L$ for all $i \in \mathbb{N}$, we have $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{Q_{f(j+1)}}^m)^L$; thus $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{Q_{f(j+1)}}^{(j+1)*})^L$. Now $(R_{Q_{f(j+1)}}^m)^L = (X_{Q_{f(j+1)}})^L \setminus (R_{Q_{f(j)}}^m)^L$, since $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{Q_{f(j)}}^i)^L$, we have $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{Q_{f(j)}}^m)^L$; hence $(R_{Q_{f(j)}}^{(j)*})^L \cap (R_{Q_{f(j+1)}}^m)^L = \emptyset$, so $(R_{Q_{f(j+2)}}^{(j+1)*})^L \cap (R_{Q_{f(j+1)}}^{(j+1)*})^L = \emptyset$. Now $(R_{Q_{f(j+2)}}^{(j+1)*})^L \cap (X_{Z_{j+1}})^L = \emptyset$, because $(X_{Z_{j+1}})^L \subseteq (X_{Z_j})^L$ and $(R_{Q_{f(j+2)}}^{(j+1)*})^L = (R_{Q_{f(j)}}^{(j)*})^L$. In Step 10, $(R_{Q_{f(j)}}^m)^L = ((R_{Q_{f(j)}}^{m-1})^L \setminus (X_{Z_j})^L) \cup (X_{S_{D_{m-1}}})^L$ and $(X_{Z_{j+1}})^L = (X_{S_{D_{m-1}}})^L$. Therefore $(R_{Q_{f(j+1)}}^m)^L \cap (X_{Z_{j+1}})^L = \emptyset$, as $(R_{Q_{f(j+1)}}^m)^L = (X_{Q_{f(j+1)}})^L \setminus (R_{Q_{f(j)}}^m)^L$; hence $(R_{Q_{f(j+1)}}^{(j+1)*})^L \cap (X_{Z_{j+1}})^L = \emptyset$. Next we show that $(X_{Q_1})^L = (R_{Q_{f(j+1)}}^{(j+1)*})^L \cup (R_{Q_{f(j+2)}}^{(j+1)*})^L \cup (X_{Z_{j+1}})^L$. By the induction hypothesis, $(X_{Q_1})^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$. By definition, $(R_{Q_{f(j+2)}}^{(j+1)*})^L = (R_{Q_{f(j)}}^{(j)*})^L$ and $(X_{Z_j})^L \setminus (X_{Z_{j+1}})^L \subseteq (R_{Q_{f(j+1)}}^{(j+1)*})^L$, and, by Lemma 5.13, $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{Q_{f(j+1)}}^m)^L = (R_{Q_{f(j+1)}}^{(j+1)*})^L$; hence $(R_{Q_{f(j+1)}}^{(j+1)*})^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup ((X_{Z_j})^L \setminus (X_{Z_{j+1}})^L)$. Therefore $(R_{Q_{f(j+1)}}^{(j+1)*})^L \cup (R_{Q_{f(j+2)}}^{(j+1)*})^L \cup (X_{Z_{j+1}})^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$; hence $(X_{Q_1})^L = (R_{Q_{f(j+1)}}^{(j+1)*})^L \cup (R_{Q_{f(j+2)}}^{(j+1)*})^L \cup (X_{Z_{j+1}})^L$. \square

Lemma 5.15. *In ITERATIVEALG, $(X_{Z_j})^L$ is structurally sound in $(Q_{f(j+1)})^L \mid (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$ and $(Q_{f(j)})^L \mid (R_{Q_{f(j+1)}}^{(j)*})^L \cup (X_{Z_j})^L$, for all $j \in \mathbb{N}$.*

Proof. We will prove this by induction on j . Base case $j = 1$: By Lemma 5.12, $(X_{Z_1})^L$ is structurally sound in both $(Q_1)^L \mid (R_{Q_2}^{(1)*})^L \cup (X_{Z_1})^L$ and $(Q_2)^L \mid (R_{Q_1}^{(1)*})^L \cup (X_{Z_1})^L$. Induction step: Suppose that $(X_{Z_j})^L$ is structurally sound in both $(Q_{f(j+1)})^L \mid (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$ and $(Q_{f(j)})^L \mid (R_{Q_{f(j+1)}}^{(j)*})^L \cup (X_{Z_j})^L$. In order to get $(R_{Q_{f(j+1)}}^{(j+1)*})^L$, $(R_{Q_{f(j+2)}}^{(j+1)*})^L$, and $(X_{Z_{j+1}})^L$, we

have to get to Step 10. In Step 10, $(R_{Q_{f(j+1)}}^{(j+1)*})^L = (R_{Q_{f(j+1)}}^m)^L$, $(R_{Q_{f(j+2)}}^{(j+1)*})^L = (R_{Q_{f(j)}}^{(j)*})^L$, and $(X_{Z_{j+1}})^L = (X_{S_{D_{m-1}}})^L$. From Step 7, we have $(X_{S_{D_{m-1}}})^L \subseteq (X_{Z_j})^L$, and, since $(X_{Z_j})^L$ is structurally sound in $(Q_{f(j+1)})^L | (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$, we have $(X_{Z_{j+1}})^L$ structurally sound in $(Q_{f(j+1)})^L | (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_{j+1}})^L$, so that $(X_{Z_{j+1}})^L$ is structurally sound in $(Q_{f(j+1)})^L | (R_{Q_{f(j+2)}}^{(j+1)*})^L \cup (X_{Z_{j+1}})^L$. By Lemma 5.14, we have $(R_{Q_{f(j+1)}}^{(j+1)*})^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup ((X_{Z_j})^L \setminus (X_{Z_{j+1}})^L)$. Now $(X_{Z_j})^L$ is structurally sound in $(Q_{f(j)})^L | (R_{Q_{f(j+1)}}^{(j)*})^L \cup (X_{Z_j})^L$; we need to show that $(X_{Z_{j+1}})^L$ is structurally sound in $(Q_{f(j+2)})^L | (R_{Q_{f(j+1)}}^{(j+1)*})^L \cup (X_{Z_{j+1}})^L$; since $(Q_{f(j+2)})^L = (Q_{f(j)})^L$ and $(R_{Q_{f(j+1)}}^{(j+1)*})^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup ((X_{Z_j})^L \setminus (X_{Z_{j+1}})^L)$, we need to show that $(X_{Z_{j+1}})^L$ is structurally sound in $(Q_{f(j)})^L | (R_{Q_{f(j+1)}}^{(j)*})^L \cup (X_{Z_j})^L$. Since $(X_{Z_{j+1}})^L = (X_{S_{D_{m-1}}})^L$ and $(X_{S_{D_{m-1}}})^L \subseteq (X_{Z_j})^L$, we have $(X_{Z_{j+1}})^L \subseteq (X_{Z_j})^L$. Since $(X_{Z_j})^L$ is structurally sound in $(Q_{f(j)})^L | (R_{Q_{f(j+1)}}^{(j)*})^L \cup (X_{Z_j})^L$, $(R_{Q_{f(j+1)}}^{(j)*})^L \cap (X_{Z_j})^L = \emptyset$, and $(X_{Z_{j+1}})^L \subseteq (X_{Z_j})^L$, $(X_{Z_{j+1}})^L$ is structurally sound in $(Q_{f(j)})^L | (R_{Q_{f(j+1)}}^{(j)*})^L \cup (X_{Z_j})^L$, because in Step 7 we found the sibling of D_{m-1} in $(Q_{f(j)})^L | (R_{Q_{f(j)}}^{m-1})^L$, which is $S_{D_{m-1}}$, so that $(X_{S_{D_{m-1}}})^L$ is structurally sound in $(Q_{f(j)})^L | (R_{Q_{f(j)}}^{m-1})^L$. Therefore $(X_{Z_{j+1}})^L$ is structurally sound in $(Q_{f(j+2)})^L | (R_{Q_{f(j+1)}}^{(j+1)*})^L \cup (X_{Z_{j+1}})^L$. \square

Proposition 5.6. *In ITERATIVEALG, $(X_{Z_{j+1}})^L$ is structurally sound in $(Q_{f(j+2)})^L | (R_{Q_{f(j+2)}}^{(j+1)*})^L \cup (X_{Z_{j+1}})^L$, for all $j \in \mathbb{N}$.*

Proof. Let $j \in \mathbb{N}$. From Lemma 5.15, $(X_{Z_j})^L$ is structurally sound in both $(Q_{f(j+1)})^L | (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$ and $(Q_{f(j)})^L | (R_{Q_{f(j+1)}}^{(j)*})^L \cup (X_{Z_j})^L$. In order to get $(R_{Q_{f(j+2)}}^{(j+1)*})^L$ and $(X_{Z_{j+1}})^L$, we must get to Step 10, and at Step 10 $(R_{Q_{f(j+2)}}^{(j+1)*})^L = (R_{Q_{f(j)}}^{(j)*})^L$ and $(X_{Z_{j+1}})^L = (X_{S_{D_{m-1}}})^L$, by definition. Hence we need to show that $(X_{S_{D_{m-1}}})^L$ is structurally sound in $(Q_{f(j)})^L | (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{S_{D_{m-1}}})^L$, as $(Q_{f(j+2)})^L = (Q_{f(j)})^L$. In Step 7 we find the sibling of D_{m-1} in $(Q_{f(j)})^L | (R_{Q_{f(j)}}^{m-1})^L$, which is $S_{D_{m-1}}$, so that $(X_{S_{D_{m-1}}})^L$ is structurally sound in $(Q_{f(j)})^L | (R_{Q_{f(j)}}^{m-1})^L$. By Lemma 5.14 and Lemma 5.13, $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{Q_{f(j)}}^{m-1})^L$, so that $(X_{S_{D_{m-1}}})^L$ is structurally sound in $(Q_{f(j)})^L | (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{S_{D_{m-1}}})^L$. \square

Lemma 5.16. *In ITERATIVEALG, $(Q_1)^L | (X_{Z_j})^L \neq (Q_2)^L | (X_{Z_j})^L$, $(Q_1)^L | (R_{Q_{f(j)}}^{(j)*})^L = (Q_2)^L | (R_{Q_{f(j)}}^{(j)*})^L$, and $(Q_1)^L | (R_{Q_{f(j+1)}}^{(j)*})^L = (Q_2)^L | (R_{Q_{f(j+1)}}^{(j)*})^L$, for all $j \in \mathbb{N}$.*

Proof. We will prove this by induction on j . Base case $j = 1$: When $j = 1$, we have $(X_{Z_1})^L$, $(R_{Q_1}^{(1)*})^L$, and $(R_{Q_2}^{(1)*})^L$, where $(X_{Z_1})^L$ is defined to be $(X_{\triangle_{Q_1}})^L \cap (X_{\diamond_{Q_2}})^L$, for some $\triangle, \diamond \in \{A, M\}$, and $(\triangle_{Q_1})^L | (X_{Z_1})^L \neq (\diamond_{Q_2})^L | (X_{Z_1})^L$. We also

have $(R_{Q_1}^{(1)*})^L \in \Xi \setminus \{(X_B)^L, (X_{Z_1})^L\}$, so that $(Q_1)^L | (R_{Q_1}^{(1)*})^L = (Q_2)^L | (R_{Q_1}^{(1)*})^L$, and $(R_{Q_2}^{(1)*})^L \in \Xi \setminus \{(X_B)^L, (X_{Z_1})^L, (R_{Q_1}^{(1)*})^L\}$, so $(Q_1)^L | (R_{Q_2}^{(1)*})^L = (Q_2)^L | (R_{Q_2}^{(1)*})^L$, as $\Xi = \{(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L, (X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L, (X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L, (X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L\}$. Induction step: Suppose that $(Q_1)^L | (X_{Z_j})^L \neq (Q_2)^L | (X_{Z_j})^L$, $(Q_1)^L | (R_{Q_{f(j)}}^{(j)*})^L = (Q_2)^L | (R_{Q_{f(j)}}^{(j)*})^L$, and $(Q_1)^L | (R_{Q_{f(j+1)}}^{(j)*})^L = (Q_2)^L | (R_{Q_{f(j+1)}}^{(j)*})^L$. The only way to get $(X_{Z_{j+1}})^L$, $(R_{Q_{f(j+1)}}^{(j+1)*})^L$, and $(R_{Q_{f(j+2)}}^{(j+1)*})^L$ is to get to Step 10, and the only way to get to Step 10 is when $(Q_1)^L | (R_{Q_{f(j)}}^m)^L \neq (Q_2)^L | (R_{Q_{f(j)}}^m)^L$, where $(R_{Q_{f(j)}}^m)^L = ((R_{Q_{f(j)}}^{m-1})^L \setminus (X_{Z_j})^L) \cup (X_{S_{D_{m-1}}})^L$, for some $m \in \mathbb{N}$ such that $m \geq i + 1$. We then must have $(Q_1)^L | (R_{Q_{f(j+1)}}^m)^L = (Q_2)^L | (R_{Q_{f(j+1)}}^m)^L$, where $(R_{Q_{f(j+1)}}^m)^L = (X_{Q_{f(j+1)}})^L \setminus (R_{Q_{f(j)}}^m)^L$, and that leads to getting $(X_{Z_{j+1}})^L = (X_{S_{D_{m-1}}})^L$, $(R_{Q_{f(j+1)}}^{(j+1)*})^L = (R_{Q_{f(j+1)}}^m)^L$, and $(R_{Q_{f(j+2)}}^{(j+1)*})^L = (R_{Q_{f(j)}}^{(j)*})^L$; hence $(Q_1)^L | (R_{Q_{f(j+1)}}^{(j+1)*})^L = (Q_2)^L | (R_{Q_{f(j+1)}}^{(j+1)*})^L$ and $(Q_1)^L | (R_{Q_{f(j+2)}}^{(j+1)*})^L = (Q_2)^L | (R_{Q_{f(j+2)}}^{(j+1)*})^L$. The only thing left to prove is that $(Q_1)^L | (X_{Z_{j+1}})^L \neq (Q_2)^L | (X_{Z_{j+1}})^L$, and to do that we claim that $((R_{Q_{f(j)}}^{m-1})^L \setminus (X_{Z_j})^L) = (R_{Q_{f(j)}}^{(j)*})^L$. From Lemma 5.13 and Lemma 5.14, $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{Q_{f(j)}}^{m-1})^L$, $(X_{Q_1})^L = (X_{Q_2})^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$, $(R_{Q_{f(j+1)}}^{(j)*})^L \cap (R_{Q_{f(j)}}^{(j)*})^L = \emptyset$, $(R_{Q_{f(j+1)}}^{(j)*})^L \cap (X_{Z_j})^L = \emptyset$, and $(R_{Q_{f(j)}}^{(j)*})^L \cap (X_{Z_j})^L = \emptyset$; hence $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{Q_{f(j)}}^{m-1})^L \setminus (X_{Z_j})^L$. Take any $x \in ((R_{Q_{f(j)}}^{m-1})^L \setminus (X_{Z_j})^L)$ then $x \in (R_{Q_{f(j)}}^{m-1})^L$ and $x \notin (X_{Z_j})^L$, so $x \in (X_{Q_1})^L$. Suppose for contradiction that $x \in (R_{Q_{f(j+1)}}^{(j)*})^L$ then $x \in (R_{Q_{f(j+1)}}^m)^L$, as, by Lemma 5.13, $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{Q_{f(j+1)}}^m)^L$. Now $(R_{Q_{f(j+1)}}^m)^L = (X_{Q_{f(j+1)}})^L \setminus (R_{Q_{f(j)}}^m)^L$, so that $x \notin (R_{Q_{f(j)}}^m)^L$, but $(R_{Q_{f(j)}}^m)^L = ((R_{Q_{f(j)}}^{m-1})^L \setminus (X_{Z_j})^L) \cup (X_{S_{D_{m-1}}})^L$; thus $x \notin ((R_{Q_{f(j)}}^{m-1})^L \setminus (X_{Z_j})^L)$ - a contradiction. Therefore $x \in (R_{Q_{f(j)}}^{(j)*})^L$, so that $((R_{Q_{f(j)}}^{m-1})^L \setminus (X_{Z_j})^L) = (R_{Q_{f(j)}}^{(j)*})^L$. By the induction hypothesis, $(Q_1)^L | (R_{Q_{f(j)}}^{(j)*})^L = (Q_2)^L | (R_{Q_{f(j)}}^{(j)*})^L$; hence $(Q_1)^L | ((R_{Q_{f(j)}}^{m-1})^L \setminus (X_{Z_j})^L) = (Q_2)^L | ((R_{Q_{f(j)}}^{m-1})^L \setminus (X_{Z_j})^L)$. Since $(X_{S_{D_{m-1}}})^L \subseteq (X_{Z_j})^L$, $((R_{Q_{f(j)}}^{m-1})^L \setminus (X_{Z_j})^L) = (R_{Q_{f(j)}}^{(j)*})^L$, and $(R_{Q_{f(j)}}^{(j)*})^L \cap (X_{Z_j})^L = \emptyset$, we have $((R_{Q_{f(j)}}^{m-1})^L \setminus (X_{Z_j})^L) \cap (X_{S_{D_{m-1}}})^L = \emptyset$. Therefore the only way that $(Q_1)^L | (R_{Q_{f(j)}}^m)^L \neq (Q_2)^L | (R_{Q_{f(j)}}^m)^L$ is when $(Q_1)^L | (X_{S_{D_{m-1}}})^L \neq (Q_2)^L | (X_{S_{D_{m-1}}})^L$, because $(X_{Z_{j+1}})^L = (X_{S_{D_{m-1}}})^L$ is structurally sound in $(Q_{f(j)})^L | (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{S_{D_{m-1}}})^L$ and $(Q_{f(j+1)})^L | (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{S_{D_{m-1}}})^L$, by Lemma 5.15 and Proposition 5.6; thus $(Q_1)^L | (X_{Z_{j+1}})^L \neq (Q_2)^L | (X_{Z_{j+1}})^L$. \square

Lemma 5.17. *In ITERATIVEALG, if $(U_{Q_{f(j+1)}}, U_{Q_{f(j)}})$ is returned then $(X_{Z_j})^L$ is structurally sound in $(Q_{f(j+1)})^L | (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$, $(Q_{f(j)})^L | (R_{Q_{f(j+1)}}^{(j)*})^L \cup (X_{Z_j})^L$,*

$$\left(Q_{f(j+1)}\right)^L \mid \left(R_{Q_{f(j+1)}}^{(j)*}\right)^L \cup \left(X_{Z_j}\right)^L, \text{ and } \left(Q_{f(j)}\right)^L \mid \left(R_{Q_{f(j)}}^{(j)*}\right)^L \cup \left(X_{Z_j}\right)^L.$$

Proof. By Lemma 5.15 and Proposition 5.6, $\left(X_{Z_j}\right)^L$ is structurally sound in $\left(Q_{f(j+1)}\right)^L \mid \left(R_{Q_{f(j+1)}}^{(j)*}\right)^L \cup \left(X_{Z_j}\right)^L$, $\left(Q_{f(j)}\right)^L \mid \left(R_{Q_{f(j+1)}}^{(j)*}\right)^L \cup \left(X_{Z_j}\right)^L$, and $\left(Q_{f(j+1)}\right)^L \mid \left(R_{Q_{f(j+1)}}^{(j)*}\right)^L \cup \left(X_{Z_j}\right)^L$. By Step 7 and Step 8, we see that $\left(X_{Z_j}\right)^L$ is structurally sound in $\left(Q_{f(j)}\right)^L \mid \left(R_{Q_{f(j)}}^{(j)*}\right)^L$. Now, by Lemma 5.14 and Lemma 5.13, $\left(R_{Q_{f(j)}}^{(j)*}\right)^L \subseteq \left(R_{Q_{f(j)}}^i\right)^L$, and, by construction and Step 8, $\left(X_{Z_j}\right)^L \subseteq \left(R_{Q_{f(j)}}^i\right)^L$, so that $\left(X_{Z_j}\right)^L$ is structurally sound in $\left(Q_{f(j)}\right)^L \mid \left(R_{Q_{f(j)}}^{(j)*}\right)^L \cup \left(X_{Z_j}\right)^L$. \square

Lemma 5.18. *In ITERATIVEALG, if $\left(U_{Q_{f(j+1)}}, U_{Q_{f(j)}}\right)$ is returned then $\left(U_{Q_{f(j+1)}}, U_{Q_{f(j)}}\right)$ is a major tree pair of \mathcal{T} .*

Proof. The only way that $\left(U_{Q_{f(j+1)}}, U_{Q_{f(j)}}\right)$ can be returned in ITERATIVEALG is when, at Step 8, $\left(X_{S_{D_i}}\right)^L = \left(X_{Z_j}\right)^L$. We need to prove that $\left(U_{Q_{f(j+1)}}\right)^L \neq \left(U_{Q_{f(j)}}\right)^L$. Now $U_{Q_{f(j+1)}}$ is defined to be the subtree of $Q_{f(j+1)}$ such that $\left(X_{U_{Q_{f(j+1)}}}\right)^L = \left(X_{Z_j}\right)^L$, so that $\left(U_{Q_{f(j+1)}}\right)^L = \left(Q_{f(j+1)}\right)^L \mid \left(X_{Z_j}\right)^L$. Also $U_{Q_{f(j)}}$ is defined to be the subtree of $Q_{f(j)}$ such that $\left(X_{U_{Q_{f(j)}}}\right)^L = \left(X_{S_{D_i}}\right)^L$, since $\left(X_{S_{D_i}}\right)^L = \left(X_{Z_j}\right)^L$, we have $\left(X_{U_{Q_{f(j)}}}\right)^L = \left(X_{Z_j}\right)^L$, so that $\left(U_{Q_{f(j)}}\right)^L = \left(Q_{f(j)}\right)^L \mid \left(X_{Z_j}\right)^L$. Therefore we need to prove that $\left(Q_{f(j+1)}\right)^L \mid \left(X_{Z_j}\right)^L \neq \left(Q_{f(j)}\right)^L \mid \left(X_{Z_j}\right)^L$, which is the case, by Lemma 5.16. However, we still need to prove that $U_{Q_{f(j+1)}}$ and $U_{Q_{f(j)}}$ exist, and the only way they can exist is when $\left(X_{Z_j}\right)^L$ is structurally sound in $(Q_1)^L$ and $(Q_2)^L$. By Lemma 5.17, $\left(X_{Z_j}\right)^L$ is structurally sound in $\left(Q_{f(j+1)}\right)^L \mid \left(R_{Q_{f(j+1)}}^{(j)*}\right)^L \cup \left(X_{Z_j}\right)^L$, $\left(Q_{f(j)}\right)^L \mid \left(R_{Q_{f(j+1)}}^{(j)*}\right)^L \cup \left(X_{Z_j}\right)^L$, $\left(Q_{f(j+1)}\right)^L \mid \left(R_{Q_{f(j+1)}}^{(j)*}\right)^L \cup \left(X_{Z_j}\right)^L$, and $\left(Q_{f(j)}\right)^L \mid \left(R_{Q_{f(j)}}^{(j)*}\right)^L \cup \left(X_{Z_j}\right)^L$. Suppose for contradiction that $\left(X_{Z_j}\right)^L$ is not structurally sound in $\left(Q_{f(j+1)}\right)^L$ then the smallest subtree in $\left(Q_{f(j+1)}\right)^L$ whose leaf set contains all of $\left(X_{Z_j}\right)^L$ contains something, call it x , from $\left(X_{Q_{f(j+1)}}\right)^L \setminus \left(X_{Z_j}\right)^L$. By Lemma 5.14, $\left(X_{Q_1}\right)^L = \left(X_{Q_2}\right)^L = \left(R_{Q_{f(j+1)}}^{(j)*}\right)^L \cup \left(R_{Q_{f(j)}}^{(j)*}\right)^L \cup \left(X_{Z_j}\right)^L$, $\left(R_{Q_{f(j+1)}}^{(j)*}\right)^L \cap \left(R_{Q_{f(j)}}^{(j)*}\right)^L = \emptyset$, $\left(R_{Q_{f(j+1)}}^{(j)*}\right)^L \cap \left(X_{Z_j}\right)^L = \emptyset$, and $\left(R_{Q_{f(j)}}^{(j)*}\right)^L \cap \left(X_{Z_j}\right)^L = \emptyset$, so $x \in \left(R_{Q_{f(j)}}^{(j)*}\right)^L$ or $x \in \left(R_{Q_{f(j+1)}}^{(j)*}\right)^L$. If $x \in \left(R_{Q_{f(j+1)}}^{(j)*}\right)^L$ then $\left(X_{Z_j}\right)^L$ is not structurally sound in $\left(Q_{f(j+1)}\right)^L \mid \left(R_{Q_{f(j+1)}}^{(j)*}\right)^L \cup \left(X_{Z_j}\right)^L$, and if $x \in \left(R_{Q_{f(j)}}^{(j)*}\right)^L$ then $\left(X_{Z_j}\right)^L$ is not structurally sound in $\left(Q_{f(j+1)}\right)^L \mid \left(R_{Q_{f(j)}}^{(j)*}\right)^L \cup \left(X_{Z_j}\right)^L$ - a contradiction. Therefore $\left(X_{Z_j}\right)^L$ is structurally sound in $\left(Q_{f(j+1)}\right)^L$. Suppose for contradiction that $\left(X_{Z_j}\right)^L$ is not structurally sound in $\left(Q_{f(j)}\right)^L$ then the smallest subtree in $\left(Q_{f(j)}\right)^L$ whose leaf set contains all of $\left(X_{Z_j}\right)^L$ contains something, call it y , from $\left(X_{Q_{f(j)}}\right)^L \setminus \left(X_{Z_j}\right)^L$. If $y \in \left(R_{Q_{f(j+1)}}^{(j)*}\right)^L$ then $\left(X_{Z_j}\right)^L$ is not structurally sound in $\left(Q_{f(j)}\right)^L \mid \left(R_{Q_{f(j+1)}}^{(j)*}\right)^L \cup \left(X_{Z_j}\right)^L$, and if $y \in \left(R_{Q_{f(j)}}^{(j)*}\right)^L$ then $\left(X_{Z_j}\right)^L$ is not structurally sound in $\left(Q_{f(j)}\right)^L \mid \left(R_{Q_{f(j)}}^{(j)*}\right)^L \cup \left(X_{Z_j}\right)^L$ - a contradiction. Therefore $\left(X_{Z_j}\right)^L$ is structurally sound in $(Q_1)^L$ and $(Q_2)^L$, so that $U_{Q_{f(j+1)}}$ and $U_{Q_{f(j)}}$ exist. Since $\left(Q_{f(j+1)}, Q_{f(j)}\right)$ is a major tree pair of \mathcal{T} , $\left(U_{Q_{f(j+1)}}, U_{Q_{f(j)}}\right)$ is a major tree pair of \mathcal{T} . \square

Proposition 5.7. *In ITERATIVEALG, if (U_{Q_1}, U_{Q_2}) is returned then $(Q_1)^L | (R_{T_1})^L \cap (X_{Q_1})^L = (Q_2)^L | (R_{T_1})^L \cap (X_{Q_2})^L$ and $(Q_1)^L | (R_{T_2})^L \cap (X_{Q_1})^L = (Q_2)^L | (R_{T_2})^L \cap (X_{Q_2})^L$ if and only if $(U_{Q_1})^L | (R_{T_1})^L \cap (X_{U_{Q_1}})^L = (U_{Q_2})^L | (R_{T_1})^L \cap (X_{U_{Q_2}})^L$ and $(U_{Q_1})^L | (R_{T_2})^L \cap (X_{U_{Q_1}})^L = (U_{Q_2})^L | (R_{T_2})^L \cap (X_{U_{Q_2}})^L$. For some switching R of \mathcal{T} with sets R_{T_1} , R_{T_2} having the following properties: $R_{T_1} \subseteq X_{T_1}$, $R_{T_2} \subseteq X_{T_2}$, $(R_{T_1})^L \cup (R_{T_2})^L = (R)^L$, and $(R_{T_1})^L \cap (R_{T_2})^L = \emptyset$.*

Proof. If (U_{Q_1}, U_{Q_2}) is returned then, by Lemma 5.18 and by Step 8, (U_{Q_1}, U_{Q_2}) is a major tree pair of \mathcal{T} with U_{Q_1} a subtree of Q_1 and U_{Q_2} a subtree of Q_2 . If $(Q_1)^L | (R_{T_1})^L \cap (X_{Q_1})^L = (Q_2)^L | (R_{T_1})^L \cap (X_{Q_2})^L$ and $(Q_1)^L | (R_{T_2})^L \cap (X_{Q_1})^L = (Q_2)^L | (R_{T_2})^L \cap (X_{Q_2})^L$ then $(U_{Q_1})^L | (R_{T_1})^L \cap (X_{U_{Q_1}})^L = (U_{Q_2})^L | (R_{T_1})^L \cap (X_{U_{Q_2}})^L$ and $(U_{Q_1})^L | (R_{T_2})^L \cap (X_{U_{Q_1}})^L = (U_{Q_2})^L | (R_{T_2})^L \cap (X_{U_{Q_2}})^L$, because $(X_{U_{Q_1}})^L \subseteq (X_{Q_1})^L$, $(X_{U_{Q_1}})^L = (X_{U_{Q_2}})^L$, and $(X_{Q_1})^L = (X_{Q_2})^L$, so that $(U_{Q_1})^L | (R_{T_1})^L \cap (X_{U_{Q_1}})^L$ is a subtree of $(Q_1)^L | (R_{T_1})^L \cap (X_{Q_1})^L$, $(U_{Q_2})^L | (R_{T_1})^L \cap (X_{U_{Q_2}})^L$ is a subtree of $(Q_2)^L | (R_{T_1})^L \cap (X_{Q_2})^L$, $(U_{Q_1})^L | (R_{T_2})^L \cap (X_{U_{Q_1}})^L$ is a subtree of $(Q_1)^L | (R_{T_2})^L \cap (X_{Q_1})^L$, and $(U_{Q_2})^L | (R_{T_2})^L \cap (X_{U_{Q_2}})^L$ is a subtree of $(Q_2)^L | (R_{T_2})^L \cap (X_{Q_2})^L$; hence, by Lemma 4.1, we get $(U_{Q_1})^L | (R_{T_1})^L \cap (X_{U_{Q_1}})^L = (U_{Q_2})^L | (R_{T_1})^L \cap (X_{U_{Q_2}})^L$ and $(U_{Q_1})^L | (R_{T_2})^L \cap (X_{U_{Q_1}})^L = (U_{Q_2})^L | (R_{T_2})^L \cap (X_{U_{Q_2}})^L$.

If $(U_{Q_1})^L | (R_{T_1})^L \cap (X_{U_{Q_1}})^L = (U_{Q_2})^L | (R_{T_1})^L \cap (X_{U_{Q_2}})^L$ and $(U_{Q_1})^L | (R_{T_2})^L \cap (X_{U_{Q_1}})^L = (U_{Q_2})^L | (R_{T_2})^L \cap (X_{U_{Q_2}})^L$ then consider $(R_{T_1})^L \cap (X_{Q_1})^L = (R_{Q_1}^{(j)*})^L \cup ((R_{T_1})^L \cap (X_{U_{Q_1}})^L)$ and $(R_{T_2})^L \cap (X_{Q_2})^L = (R_{Q_2}^{(j)*})^L \cup ((R_{T_2})^L \cap (X_{U_{Q_2}})^L)$. Now R is still a switching, by Lemma 5.14. Since $(R_{T_1})^L \cap (X_{U_{Q_1}})^L \subseteq (X_{Z_j})^L$, $(R_{T_2})^L \cap (X_{U_{Q_2}})^L \subseteq (X_{Z_j})^L$, and, by Lemma 5.17, $(X_{Z_j})^L$ is structurally sound in $(Q_1)^L | (R_{Q_1}^{(j)*})^L \cup (X_{Z_j})^L$, $(Q_1)^L | (R_{Q_2}^{(j)*})^L \cup (X_{Z_j})^L$, $(Q_2)^L | (R_{Q_1}^{(j)*})^L \cup (X_{Z_j})^L$, and $(Q_2)^L | (R_{Q_2}^{(j)*})^L \cup (X_{Z_j})^L$, we have $(R_{T_1})^L \cap (X_{U_{Q_1}})^L$ structurally sound in $(Q_1)^L | (R_{Q_1}^{(j)*})^L \cup ((R_{T_1})^L \cap (X_{U_{Q_1}})^L)$ and $(Q_2)^L | (R_{Q_1}^{(j)*})^L \cup ((R_{T_1})^L \cap (X_{U_{Q_1}})^L)$, and $(R_{T_2})^L \cap (X_{U_{Q_2}})^L$ structurally sound in $(Q_1)^L | (R_{Q_2}^{(j)*})^L \cup ((R_{T_2})^L \cap (X_{U_{Q_2}})^L)$ and $(Q_2)^L | (R_{Q_2}^{(j)*})^L \cup ((R_{T_2})^L \cap (X_{U_{Q_2}})^L)$. Therefore the two maximal subtrees of $(Q_1)^L | (R_{Q_1}^{(j)*})^L \cup ((R_{T_1})^L \cap (X_{U_{Q_1}})^L)$ are $(Q_1)^L | (R_{Q_1}^{(j)*})^L$ and $(Q_1)^L | ((R_{T_1})^L \cap (X_{U_{Q_1}})^L) = (U_{Q_1})^L | ((R_{T_1})^L \cap (X_{U_{Q_1}})^L)$, and the two maximal subtrees of $(Q_2)^L | (R_{Q_1}^{(j)*})^L \cup ((R_{T_1})^L \cap (X_{U_{Q_1}})^L)$ are $(Q_2)^L | (R_{Q_1}^{(j)*})^L$ and $(Q_2)^L | ((R_{T_1})^L \cap (X_{U_{Q_1}})^L) = (U_{Q_2})^L | ((R_{T_1})^L \cap (X_{U_{Q_2}})^L)$. By Lemma 5.16 and hypothesis, $(Q_1)^L | (R_{Q_1}^{(j)*})^L = (Q_2)^L | (R_{Q_1}^{(j)*})^L$ and $(U_{Q_1})^L | ((R_{T_1})^L \cap (X_{U_{Q_1}})^L) = (U_{Q_2})^L | ((R_{T_1})^L \cap (X_{U_{Q_2}})^L)$, so that $(Q_1)^L | (R_{T_1})^L \cap (X_{Q_1})^L = (Q_2)^L | (R_{T_1})^L \cap (X_{Q_2})^L$. Similarly, $(Q_1)^L | (R_{T_2})^L \cap (X_{Q_1})^L = (Q_2)^L | (R_{T_2})^L \cap (X_{Q_2})^L$. \square

Lemma 5.19. *Let T_1 and T_2 be two trees on the same leaf set X . Let $X_1 \subseteq X$, and let X_1 be*

structurally sound in T_i , where $i \in \{1, 2\}$. If X_1 is not structurally sound in T_j , where $j \in \{1, 2\} \setminus \{i\}$, then $T_1 \neq T_2$.

Proof. Without loss of generality, suppose that X_1 is structurally sound in T_1 and X_1 is not structurally sound in T_2 then we have a subtree of T_1 whose leaf set is X_1 , yet no such subtree exists in T_2 ; hence $T_1 \neq T_2$. \square

Lemma 5.20. *Let \mathcal{T} be the basic⁰ representation of a basic cyclic pair. Let R be any switching of \mathcal{T} , and let \mathcal{T}_1 and \mathcal{T}_2 be \mathcal{T} 's two maximal subtrees. Let $R_{\mathcal{T}_1}$ and $R_{\mathcal{T}_2}$ be two sets such that $R_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $R_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, $(R_{\mathcal{T}_1})^L \cup (R_{\mathcal{T}_2})^L = (R)^L$, and $(R_{\mathcal{T}_1})^L \cap (R_{\mathcal{T}_2})^L = \emptyset$. Let (Q_1, Q_2) be a tree pair of \mathcal{T} , and let $(X_Z)^L$ be a subset of $(X_{Q_1})^L$ such that $(Q_1)^L | (X_Z)^L \neq (Q_2)^L | (X_Z)^L$. If $(X_Z)^L \subseteq (R_{\mathcal{T}_i})^L$, for $i \in \{1, 2\}$, then $(Q_1)^L | (R_{\mathcal{T}_i})^L \cap (X_{Q_1})^L \neq (Q_2)^L | (R_{\mathcal{T}_i})^L \cap (X_{Q_2})^L$.*

Proof. Suppose that $(X_Z)^L \subseteq (R_{\mathcal{T}_i})^L$, for $i \in \{1, 2\}$, and suppose for contradiction that $(Q_1)^L | (R_{\mathcal{T}_i})^L \cap (X_{Q_1})^L = (Q_2)^L | (R_{\mathcal{T}_i})^L \cap (X_{Q_2})^L$. We must have $((Q_1)^L | (R_{\mathcal{T}_i})^L \cap (X_{Q_1})^L) | (X_Z)^L = ((Q_2)^L | (R_{\mathcal{T}_i})^L \cap (X_{Q_2})^L) | (X_Z)^L$, but $((Q_1)^L | (R_{\mathcal{T}_i})^L \cap (X_{Q_1})^L) | (X_Z)^L = (Q_1)^L | (X_Z)^L$ and $((Q_2)^L | (R_{\mathcal{T}_i})^L \cap (X_{Q_2})^L) | (X_Z)^L = (Q_2)^L | (X_Z)^L$; hence $(Q_1)^L | (X_Z)^L = (Q_2)^L | (X_Z)^L$ - a contradiction. Therefore $(Q_1)^L | (R_{\mathcal{T}_i})^L \cap (X_{Q_1})^L \neq (Q_2)^L | (R_{\mathcal{T}_i})^L \cap (X_{Q_2})^L$. \square

Proposition 5.8. *Let \mathcal{T} be the basic⁰ representation of a basic cyclic pair. Let R be any switching of \mathcal{T} , and let \mathcal{T}_1 and \mathcal{T}_2 be \mathcal{T} 's two maximal subtrees. Let $R_{\mathcal{T}_1}$ and $R_{\mathcal{T}_2}$ be two sets such that $R_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $R_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, $(R_{\mathcal{T}_1})^L \cup (R_{\mathcal{T}_2})^L = (R)^L$, and $(R_{\mathcal{T}_1})^L \cap (R_{\mathcal{T}_2})^L = \emptyset$. In ITERATIVEALG, if $(R_{\mathcal{T}_p})^L \cap (R_{Q_{f(j)}}^{(j)*})^L \neq \emptyset$, $(R_{\mathcal{T}_p})^L \cap (R_{Q_{f(j+1)}}^{(j)*})^L \neq \emptyset$, and $(R_{\mathcal{T}_p})^L \cap (X_{Z_j})^L \neq \emptyset$, for $p \in \{1, 2\}$ and $j \in \mathbb{N}$, then $(Q_1)^L | (R_{\mathcal{T}_p})^L \cap (X_{Q_1})^L \neq (Q_2)^L | (R_{\mathcal{T}_p})^L \cap (X_{Q_2})^L$.*

Proof. We will prove this by induction on j . Base case $j = 1$: By definition, $(R_{Q_1}^{(1)*})^L \in \Xi \setminus \{(X_B)^L, (X_{Z_1})^L\}$ and $(R_{Q_2}^{(1)*})^L \in \Xi \setminus \{(X_B)^L, (X_{Z_1})^L, (R_{Q_1}^{(1)*})^L\}$, where $\Xi = \{(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L, (X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L, (X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L, (X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L\}$. Without loss of generality, suppose that $(R_{\mathcal{T}_1})^L \cap (R_{Q_1}^{(1)*})^L \neq \emptyset$, $(R_{\mathcal{T}_1})^L \cap (R_{Q_2}^{(1)*})^L \neq \emptyset$, and $(R_{\mathcal{T}_1})^L \cap (X_{Z_1})^L \neq \emptyset$ then, by Lemma 5.11, $(Q_1)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L \neq (Q_2)^L | (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$. Induction hypothesis: If $(R_{\mathcal{T}_p})^L \cap (R_{Q_{f(j)}}^{(j)*})^L \neq \emptyset$, $(R_{\mathcal{T}_p})^L \cap (R_{Q_{f(j+1)}}^{(j)*})^L \neq \emptyset$, and $(R_{\mathcal{T}_p})^L \cap (X_{Z_j})^L \neq \emptyset$ then $(Q_1)^L | (R_{\mathcal{T}_p})^L \cap (X_{Q_1})^L \neq (Q_2)^L | (R_{\mathcal{T}_p})^L \cap (X_{Q_2})^L$. Without loss of generality, suppose that $(R_{\mathcal{T}_1})^L \cap (R_{Q_{f(j+1)}}^{(j+1)*})^L \neq \emptyset$, $(R_{\mathcal{T}_1})^L \cap (R_{Q_{f(j+2)}}^{(j+1)*})^L \neq \emptyset$, and $(R_{\mathcal{T}_1})^L \cap (X_{Z_{j+1}})^L \neq \emptyset$. Now, by Step 10, $(R_{Q_{f(j+1)}}^{(j+1)*})^L = (R_{Q_{f(j+1)}}^m)^L$, $(R_{Q_{f(j+2)}}^{(j+1)*})^L = (R_{Q_{f(j)}}^{(j)*})^L$, and $(X_{Z_{j+1}})^L = (X_{S_{D_{m-1}}})^L$, so $(R_{\mathcal{T}_1})^L \cap (R_{Q_{f(j)}}^{(j)*})^L \neq \emptyset$. By Step 7, $(X_{S_{D_{m-1}}})^L \subseteq (X_{Z_j})^L$, so we have $(R_{\mathcal{T}_1})^L \cap (X_{Z_j})^L \neq \emptyset$. If

$(R_{T_1})^L \cap (R_{Q_{f(j+1)}}^{(j)*})^L \neq \emptyset$ then we are done, so suppose that $(R_{T_1})^L \cap (R_{Q_{f(j+1)}}^{(j)*})^L = \emptyset$. By Lemma 5.14, $(R_{Q_{f(j+1)}}^{(j+1)*})^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup \left((X_{Z_j})^L \setminus (X_{Z_{j+1}})^L \right)$; hence $(R_{T_1})^L \cap \left((X_{Z_j})^L \setminus (X_{Z_{j+1}})^L \right) \neq \emptyset$. Since R is a switching of \mathcal{T} and $(R_{T_1})^L \cap (R_{Q_{f(j+1)}}^{(j)*})^L = \emptyset$, we have $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{T_2})^L$. By Lemma 5.20, if $(X_{Z_j})^L \subseteq (R_{T_1})^L$ then $(Q_1)^L | (R_{T_1})^L \cap (X_{Q_1})^L \neq (Q_2)^L | (R_{T_1})^L \cap (X_{Q_2})^L$, because, by Lemma 5.16, $(Q_1)^L | (X_{Z_j})^L \neq (Q_2)^L | (X_{Z_j})^L$. Since R is a switching of \mathcal{T} , we have $(R_{T_2})^L \cap (X_{Z_j})^L \neq \emptyset$. If $(R_{T_2})^L \cap (R_{Q_{f(j)}}^{(j)*})^L \neq \emptyset$ then we are done, so suppose that $(R_{T_2})^L \cap (R_{Q_{f(j)}}^{(j)*})^L = \emptyset$ then $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{T_1})^L$. Since $(R_{T_1})^L \cap \left((X_{Z_j})^L \setminus (X_{Z_{j+1}})^L \right) \neq \emptyset$ and $(R_{T_1})^L \cap (X_{Z_{j+1}})^L \neq \emptyset$, consider $\left[(R_{T_1})^L \cap \left((X_{Z_j})^L \setminus (X_{Z_{j+1}})^L \right) \right] \cup \left((R_{T_1})^L \cap (X_{Z_{j+1}})^L \right)$. We see that $\left[(R_{T_1})^L \cap \left((X_{Z_j})^L \setminus (X_{Z_{j+1}})^L \right) \right] \cup \left((R_{T_1})^L \cap (X_{Z_{j+1}})^L \right)$ is structurally sound in $(Q_{f(j+1)})^L | (R_{T_1})^L \cap (X_{Q_{f(j+1)}})^L$, because, by Lemma 5.15, $(X_{Z_j})^L$ is structurally sound in $(Q_{f(j+1)})^L | (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$ and $(X_{Z_{j+1}})^L \subseteq (X_{Z_j})^L$. By construction, $\left[(R_{T_1})^L \cap \left((X_{Z_j})^L \setminus (X_{Z_{j+1}})^L \right) \right] \cup \left((R_{T_1})^L \cap (X_{Z_{j+1}})^L \right)$ is not structurally sound in $(Q_{f(j)})^L | (R_{T_1})^L \cap (X_{Q_{f(j)}})^L$, because the smallest subtree in $(Q_{f(j)})^L | (R_{T_1})^L \cap (X_{Q_{f(j)}})^L$ containing all of $\left[(R_{T_1})^L \cap \left((X_{Z_j})^L \setminus (X_{Z_{j+1}})^L \right) \right] \cup \left((R_{T_1})^L \cap (X_{Z_{j+1}})^L \right)$ must contain some of $(X_{S_{Z_j}})^L$ and $(X_{S_{Z_j}})^L \subseteq (R_{Q_{f(j)}}^{(j)*})^L$. Therefore $(Q_1)^L | (R_{T_1})^L \cap (X_{Q_1})^L \neq (Q_2)^L | (R_{T_1})^L \cap (X_{Q_2})^L$. \square

Theorem 5.5. *Let \mathcal{T} be the basic⁰ representation of a basic cyclic pair. Let R be any switching of \mathcal{T} , and let T_1 and T_2 be \mathcal{T} 's two maximal subtrees. Let R_{T_1} and R_{T_2} be two sets such that $R_{T_1} \subseteq X_{T_1}$, $R_{T_2} \subseteq X_{T_2}$, $(R_{T_1})^L \cup (R_{T_2})^L = (R)^L$, and $(R_{T_1})^L \cap (R_{T_2})^L = \emptyset$. In ITERATIVEALG, if the statement “does not display a tree twice” is returned, upon input of a sterile major tree pair (Q_1, Q_2) , then $(Q_1)^L | (R_{T_p})^L \cap (X_{Q_1})^L \neq (Q_2)^L | (R_{T_p})^L \cap (X_{Q_2})^L$, for some $p \in \{1, 2\}$.*

Proof. Suppose that the statement “does not display a tree twice” is returned. There are several ways in which that can occur. First let us look at $(R_{T_p})^L \cap (R_{Q_{f(j)}}^{(j)*})^L$, $(R_{T_p})^L \cap (R_{Q_{f(j+1)}}^{(j)*})^L$, and $(R_{T_p})^L \cap (X_{Z_j})^L$. By Proposition 5.8, if $(R_{T_p})^L \cap (R_{Q_{f(j)}}^{(j)*})^L \neq \emptyset$, $(R_{T_p})^L \cap (R_{Q_{f(j+1)}}^{(j)*})^L \neq \emptyset$, and $(R_{T_p})^L \cap (X_{Z_j})^L \neq \emptyset$ then $(Q_1)^L | (R_{T_p})^L \cap (X_{Q_1})^L \neq (Q_2)^L | (R_{T_p})^L \cap (X_{Q_2})^L$. Now, by Lemma 5.20 and Lemma 5.16, if $(X_{Z_j})^L \subseteq (R_{T_p})^L$ then $(Q_1)^L | (R_{T_p})^L \cap (X_{Q_1})^L \neq (Q_2)^L | (R_{T_p})^L \cap (X_{Q_2})^L$, so, without loss of generality, suppose that $(R_{T_1})^L \cap (R_{Q_{f(j)}}^{(j)*})^L = \emptyset$, $(R_{T_1})^L \cap (R_{Q_{f(j+1)}}^{(j)*})^L \neq \emptyset$, and $(R_{T_1})^L \cap (X_{Z_j})^L \neq \emptyset$ then $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{T_2})^L$; hence $(R_{T_2})^L \cap (R_{Q_{f(j)}}^{(j)*})^L \neq \emptyset$, $(R_{T_2})^L \cap (R_{Q_{f(j+1)}}^{(j)*})^L = \emptyset$, and $(R_{T_2})^L \cap (X_{Z_j})^L \neq \emptyset$, which implies that $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{T_1})^L$. Therefore $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{T_2})^L$ and $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{T_1})^L$. Returning to the implications of the algorithm returning “does not display a tree twice,” suppose that in Step 5 $(Q_1)^L | (R_{Q_{f(j+1)}}^{(j)*})^L \neq$

$(Q_2)^L \mid (R_{Q_{f(j+1)}}^i)^L$, where $i, j \in \mathbb{N}$ and $(R_{Q_{f(j+1)}}^i)^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup ((X_{D_{i-1}}) \setminus (X_{S_{Z_j}})^L)$. Now $((X_{D_{i-1}}) \setminus (X_{S_{Z_j}})^L) \subseteq (X_{Z_j})^L$, and if $((X_{D_{i-1}}) \setminus (X_{S_{Z_j}})^L) \subseteq (R_{\mathcal{T}_1})^L$ then, by Lemma 5.20, $(Q_1)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L \neq (Q_2)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$. Suppose that $((X_{D_{i-1}}) \setminus (X_{S_{Z_j}})^L) \not\subseteq (R_{\mathcal{T}_1})^L$ then, since R is a switching, $(R_{\mathcal{T}_2})^L \cap ((X_{D_{i-1}}) \setminus (X_{S_{Z_j}})^L) \neq \emptyset$. Since $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{\mathcal{T}_2})^L$ and $((X_{D_{i-1}}) \setminus (X_{S_{Z_j}})^L) \neq \emptyset$, $(X_{S_{Z_j}})^L$ is not structurally sound in $(Q_{f(j)})^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_{f(j)}})^L$, yet $(X_{S_{Z_j}})^L$ is structurally sound in $(Q_{f(j+1)})^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_{f(j+1)}})^L$, because, by Lemma 5.15, $(X_{Z_j})^L$ is structurally sound in $(Q_{f(j+1)})^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_{f(j+1)}})^L$ and $((X_{D_{i-1}}) \setminus (X_{S_{Z_j}})^L) \subseteq (X_{Z_j})^L$. Therefore, by Lemma 5.19, $(Q_1)^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_1})^L \neq (Q_2)^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L$, as $(X_{S_{Z_j}})^L \subseteq (R_{Q_{f(j)}}^{(j)*})^L$.

Suppose that in Step 7 $(X_{S_{D_i}})^L \not\subseteq (X_{Z_j})^L$ then there exists $x \in (X_{S_{D_i}})^L$ such that $x \notin (X_{Z_j})^L$. By Lemma 5.14, $(X_{Q_1})^L = (X_{Q_2})^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$, $(R_{Q_{f(j+1)}}^{(j)*})^L \cap (R_{Q_{f(j)}}^{(j)*})^L = \emptyset$, $(R_{Q_{f(j+1)}}^{(j)*})^L \cap (X_{Z_j})^L = \emptyset$, and $(R_{Q_{f(j)}}^{(j)*})^L \cap (X_{Z_j})^L = \emptyset$; hence $x \in (R_{\mathcal{T}_2})^L$ because $(R_{Q_{f(j)}}^{(j)*})^L \subseteq (R_{\mathcal{T}_2})^L$. Since $(R_{\mathcal{T}_2})^L \cap (X_{Z_j})^L \neq \emptyset$, we see that $(X_{S_{Z_j}})^L \cup ((R_{\mathcal{T}_2})^L \cap (X_{Z_j})^L)$ is structurally sound in $(Q_{f(j+1)})^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_{f(j+1)}})^L$, as $(X_{S_{Z_j}})^L \subseteq (R_{Q_{f(j)}}^{(j)*})^L$, yet $(X_{S_{Z_j}})^L \cup ((R_{\mathcal{T}_2})^L \cap (X_{Z_j})^L)$ is not structurally sound in $(Q_{f(j)})^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_{f(j)}})^L$, because the smallest subtree in $(Q_{f(j)})^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_{f(j)}})^L$ containing all of $(X_{S_{Z_j}})^L \cup ((R_{\mathcal{T}_2})^L \cap (X_{Z_j})^L)$ must contain x , and $x \notin (X_{S_{Z_j}})^L$ by construction. Therefore, by Lemma 5.19, $(Q_1)^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_1})^L \neq (Q_2)^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L$.

Suppose that in Step 9 $(Q_1)^L \mid (R_{Q_{f(j)}}^{i+2q-1})^L = (Q_2)^L \mid (R_{Q_{f(j)}}^{i+2q-1})^L$, $(Q_1)^L \mid (R_{Q_{f(j+1)}}^{i+2q-1})^L \neq (Q_2)^L \mid (R_{Q_{f(j+1)}}^{i+2q-1})^L$, and $(Q_1)^L \mid (R_{Q_{f(j+1)}}^{i+2q})^L \neq (Q_2)^L \mid (R_{Q_{f(j+1)}}^{i+2q})^L$, where $(R_{Q_{f(j)}}^{i+2q-1})^L = ((R_{Q_{f(j)}}^{i+2(q-1)})^L \setminus (X_{Z_j})^L) \cup (X_{S_{D_{i+2(q-1)}}})^L$, $(R_{Q_{f(j+1)}}^{i+2q-1})^L = (X_{Q_{f(j+1)}})^L \setminus (R_{Q_{f(j)}}^{i+2q-1})^L$, and $(R_{Q_{f(j+1)}}^{i+2q})^L = (R_{Q_{f(j+1)}}^{i+2(q-1)})^L \cup (X_{S_{D_{i+2(q-1)}}})^L$, for some $q \in \mathbb{N}$. If $(R_{\mathcal{T}_2})^L \cap ((X_{Z_j})^L \setminus (X_{S_{D_{i+2(q-1)}}})^L) \neq \emptyset$ and $(R_{\mathcal{T}_2})^L \cap (X_{S_{D_{i+2(q-1)}}})^L \neq \emptyset$ then $(Q_1)^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_1})^L \neq (Q_2)^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L$, because $(X_{Z_j})^L$ is structurally sound in $(Q_{f(j+1)})^L \mid (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$ but, since $(X_{S_{D_{i+2(q-1)}}})^L \subseteq (X_{Z_j})^L$, $S_{D_{i+2(q-1)}}$ is the sibling of $D_{i+2(q-1)}$, and $(X_{D_{i+2(q-1)}})^L \subseteq (R_{Q_{f(j)}}^{(j)*})^L$, $(X_{Z_j})^L$ is not structurally sound in $(Q_{f(j)})^L \mid (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$, so that $(Q_{f(j)})^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_{f(j)}})^L$ contains the triple $xz|y$, where $x \in (R_{\mathcal{T}_2})^L \cap (X_{S_{D_{i+2(q-1)}}})^L$, $y \in (R_{\mathcal{T}_2})^L \cap ((X_{Z_j})^L \setminus (X_{S_{D_{i+2(q-1)}}})^L)$, and $z \in (X_{D_{i+2(q-1)}})^L$, whilst $(Q_{f(j+1)})^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_{f(j+1)}})^L$ does not, so, by Lemma 1.2, $(Q_1)^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_1})^L \neq$

$(Q_2)^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L$. If $(R_{\mathcal{T}_2})^L \cap \left((X_{Z_j})^L \setminus (X_{S_{D_{i+2(q-1)}}})^L \right) = \emptyset$ then, since R is a switching, $\left((X_{Z_j})^L \setminus (X_{S_{D_{i+2(q-1)}}})^L \right) \subseteq (R_{\mathcal{T}_1})^L$. This means that $(R_{Q_{f(j+1)}}^{i+2q-1})^L \subseteq (R_{\mathcal{T}_1})^L$, so, by Lemma 5.20, $(Q_1)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L \neq (Q_2)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$. If $(R_{\mathcal{T}_2})^L \cap (X_{S_{D_{i+2(q-1)}}})^L = \emptyset$ then $(X_{S_{D_{i+2(q-1)}}})^L \subseteq (R_{\mathcal{T}_1})^L$. Consider $(R_{Q_{f(j+1)}}^{i+2(q-1)})^L$ for $q = 1$ we have $(R_{Q_{f(j+1)}}^{i+2(q-1)})^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup \left((X_{D_{i-1}}) \setminus (X_{S_{Z_j}}) \right)^L$, by Step 5, and in this case we have $(R_{Q_{f(j+1)}}^{i+2(q-1)})^L \subseteq (R_{\mathcal{T}_1})^L$, because $(R_{\mathcal{T}_2})^L \cap \left((X_{D_{i-1}}) \setminus (X_{S_{Z_j}}) \right)^L = \emptyset$, from the first paragraph, which means that $(R_{Q_{f(j+1)}}^{i+2q})^L \subseteq (R_{\mathcal{T}_1})^L$; hence for $q = 1$ we get $(Q_1)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L \neq (Q_2)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$, by Lemma 5.20. For $q > 1$, we see that $(R_{Q_{f(j+1)}}^{i+2(q-1)})^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup \left((X_{D_{i-1}}) \setminus (X_{S_{Z_j}}) \right)^L \cup (X_{S_{D_i}})^L \cup \dots \cup (X_{S_{D_{i+2(q-2)}}})^L$. Suppose that $(R_{\mathcal{T}_2})^L \cap (X_{S_{D_\alpha}})^L \neq \emptyset$, for some $\alpha \in \{i, \dots, i+2(q-2)\}$. If $(R_{\mathcal{T}_2})^L \cap \left((X_{Z_j})^L \setminus (X_{S_{D_\alpha}})^L \right) \neq \emptyset$ then $(Q_1)^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_1})^L \neq (Q_2)^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L$, for the same reasons when $\alpha = i+2(q-1)$. Suppose that $(R_{\mathcal{T}_2})^L \cap \left((X_{Z_j})^L \setminus (X_{S_{D_\alpha}})^L \right) = \emptyset$, but this means that $\left((X_{Z_j})^L \setminus (X_{S_{D_\alpha}})^L \right) \subseteq (R_{\mathcal{T}_1})^L$. Now we must have $(Q_1)^L \mid (R_{Q_{f(j)}}^{\alpha+1})^L = (Q_2)^L \mid (R_{Q_{f(j)}}^{\alpha+1})^L$ and $(Q_1)^L \mid (R_{Q_{f(j+1)}}^{\alpha+1})^L \neq (Q_2)^L \mid (R_{Q_{f(j+1)}}^{\alpha+1})^L$, where $(R_{Q_{f(j)}}^{\alpha+1})^L = \left((R_{Q_{f(j)}}^\alpha) \setminus (X_{Z_j}) \right)^L \cup (X_{S_{D_\alpha}})^L$ and $(R_{Q_{f(j+1)}}^{\alpha+1})^L = (X_{Q_{f(j+1)}})^L \setminus (R_{Q_{f(j)}}^{\alpha+1})^L$. We see that $(R_{Q_{f(j)}}^{\alpha+1})^L = (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{S_{D_\alpha}})^L$ and $(R_{Q_{f(j+1)}}^{\alpha+1})^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup \left((X_{Z_j})^L \setminus (X_{S_{D_\alpha}})^L \right)$, so that $(R_{Q_{f(j+1)}}^{\alpha+1})^L \subseteq (R_{\mathcal{T}_1})^L$; hence, by Lemma 5.20, $(Q_1)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L \neq (Q_2)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$. Suppose that $(R_{\mathcal{T}_2})^L \cap (X_{S_{D_\alpha}})^L = \emptyset$, for all $\alpha \in \{i, \dots, i+2(q-2)\}$, then $(R_{Q_{f(j+1)}}^{i+2(q-1)})^L \subseteq (R_{\mathcal{T}_1})^L$, but this means that $(R_{Q_{f(j+1)}}^{i+2q})^L \subseteq (R_{\mathcal{T}_1})^L$, as $(X_{S_{D_{i+2(q-1)}}})^L \subseteq (R_{\mathcal{T}_1})^L$; hence, by Lemma 5.20, $(Q_1)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L \neq (Q_2)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$.

Suppose that in Step 10 we have $(Q_1)^L \mid (R_{Q_{f(j)}}^m)^L \neq (Q_2)^L \mid (R_{Q_{f(j)}}^m)^L$ and $(Q_1)^L \mid (R_{Q_{f(j+1)}}^m)^L \neq (Q_2)^L \mid (R_{Q_{f(j+1)}}^m)^L$, for some $m \in \mathbb{N}$, where $(R_{Q_{f(j)}}^m)^L = \left((R_{Q_{f(j)}}^{m-1})^L \setminus (X_{Z_j})^L \right) \cup (X_{S_{D_{m-1}}})^L$ and $(R_{Q_{f(j+1)}}^m)^L = (X_{Q_{f(j+1)}})^L \setminus (R_{Q_{f(j)}}^m)^L$. From Lemma 5.16, $(Q_1)^L \mid (X_{S_{D_{m-1}}})^L \neq (Q_2)^L \mid (X_{S_{D_{m-1}}})^L$, as $(X_{S_{D_{m-1}}})^L = (X_{Z_{j+1}})^L$. We also see that $(R_{Q_{f(j)}}^m)^L = (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{S_{D_{m-1}}})^L$ and $(R_{Q_{f(j+1)}}^m)^L = (R_{Q_{f(j+1)}}^{(j)*})^L \cup \left((X_{Z_j})^L \setminus (X_{S_{D_{m-1}}})^L \right)$. If $(R_{\mathcal{T}_2})^L \cap (X_{S_{D_{m-1}}})^L \neq \emptyset$ and $(R_{\mathcal{T}_2})^L \cap \left((X_{Z_j})^L \setminus (X_{S_{D_{m-1}}})^L \right) \neq \emptyset$ then, as in the previous paragraph, $(Q_1)^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_1})^L \neq (Q_2)^L \mid (R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L$. If $(R_{\mathcal{T}_2})^L \cap (X_{S_{D_{m-1}}})^L = \emptyset$ then, since R is a switching, $(X_{S_{D_{m-1}}})^L \subseteq (R_{\mathcal{T}_1})^L$. Since $(Q_1)^L \mid (X_{S_{D_{m-1}}})^L \neq (Q_2)^L \mid (X_{S_{D_{m-1}}})^L$, we have, by Lemma 5.20, $(Q_1)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L \neq (Q_2)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$. If $(R_{\mathcal{T}_2})^L \cap \left((X_{Z_j})^L \setminus (X_{S_{D_{m-1}}})^L \right) = \emptyset$ then $\left((X_{Z_j})^L \setminus (X_{S_{D_{m-1}}})^L \right) \subseteq (R_{\mathcal{T}_1})^L$. Since $(R_{Q_{f(j+1)}}^{(j)*})^L \subseteq (R_{\mathcal{T}_1})^L$, we have $(R_{Q_{f(j+1)}}^m)^L \subseteq (R_{\mathcal{T}_1})^L$, which implies, by Lemma 5.20,

$$(Q_1)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L \neq (Q_2)^L \mid (R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L. \quad \square$$

5.3.3.4 Running-time of the iterative algorithm

Lemma 5.21. *In ITERATIVEALG, S_{Z_j} exists in Step 3 and S_{D_i} exists in Step 7, for $i, j \in \mathbb{N}$. Moreover, $(X_{S_{Z_j}})^L \subseteq (R_{Q_{f(j)}}^{(j)*})^L$.*

Proof. It can be shown that $(R_{Q_{f(j)}}^{(1)*})^L \subseteq (R_{Q_{f(j)}}^{(j)*})^L$, for $j \in \mathbb{N}$. Since $(R_{Q_{f(j)}}^{(1)*})^L \neq \emptyset$ by definition, we have $(R_{Q_{f(j)}}^{(j)*})^L \neq \emptyset$. We also have $(R_{Q_{f(j)}}^{(j)*})^L \cap (X_{Z_j})^L = \emptyset$, by Lemma 5.14. By Lemma 5.15, $(X_{Z_j})^L$ is structurally sound in $(Q_{f(j+1)})^L \mid (R_{Q_{f(j)}}^{(j)*})^L \cup (X_{Z_j})^L$, so that S_{Z_j} exists in Step 3, and $(X_{S_{Z_j}})^L \subseteq (R_{Q_{f(j)}}^{(j)*})^L$. We must have $(X_{S_{Z_j}})^L \neq \emptyset$, since S_{Z_j} exists, so in Step 4 D_{i-1} exists. In Step 5 we must have $(Q_1)^L \mid (R_{Q_{f(j+1)}}^i)^L \neq (Q_2)^L \mid (R_{Q_{f(j+1)}}^i)^L$, otherwise the algorithm would stop. In Step 6 D_i exists, so that $(X_{S_{Z_j}})^L$ is structurally sound in $(Q_{f(j)})^L \mid (R_{Q_{f(j)}}^i)^L$. If $(R_{Q_{f(j)}}^i)^L \cap (X_{Z_j})^L = \emptyset$ then $(R_{Q_{f(j)}}^i)^L = (R_{Q_{f(j)}}^{(j)*})^L$. By Lemma 5.16, $(Q_1)^L \mid (R_{Q_{f(j)}}^{(j)*})^L = (Q_2)^L \mid (R_{Q_{f(j)}}^{(j)*})^L$, so that if $(R_{Q_{f(j)}}^i)^L \cap (X_{Z_j})^L = \emptyset$ then $(Q_1)^L \mid (R_{Q_{f(j)}}^i)^L = (Q_2)^L \mid (R_{Q_{f(j)}}^i)^L$; hence we never get to Step 7 if $(R_{Q_{f(j)}}^i)^L \cap (X_{Z_j})^L = \emptyset$. Therefore $(R_{Q_{f(j)}}^i)^L \cap (X_{Z_j})^L \neq \emptyset$; thus in Step 7 S_{D_i} exists. \square

Lemma 5.22. *In ITERATIVEALG, if in Step 9 we go back to Step 6, replacing i with $i + 2$, then $S_{D_{i+2}}$ exists in Step 7.*

Proof. As in the proof of Lemma 5.21, we need to prove that $(R_{Q_{f(j)}}^{i+2})^L \cap (X_{Z_j})^L \neq \emptyset$. Suppose for contradiction that $(R_{Q_{f(j)}}^{i+2})^L \cap (X_{Z_j})^L = \emptyset$ then $(X_{Z_j})^L \subseteq (R_{Q_{f(j+1)}}^{i+2})^L$, but, by Lemma 5.20 and Lemma 5.16, this implies that $(Q_1)^L \mid (R_{Q_{f(j+1)}}^{i+2})^L \neq (Q_2)^L \mid (R_{Q_{f(j+1)}}^{i+2})^L$, which is impossible, because in order to get $(R_{Q_{f(j)}}^{i+2})^L$ we have to have had $(Q_1)^L \mid (R_{Q_{f(j+1)}}^{i+2})^L = (Q_2)^L \mid (R_{Q_{f(j+1)}}^{i+2})^L$. Therefore $(R_{Q_{f(j)}}^{i+2})^L \cap (X_{Z_j})^L \neq \emptyset$, so that $S_{D_{i+2}}$ exists in Step 7. \square

Proposition 5.9. *ITERATIVEALG runs in polynomial-time.*

Proof. First, the checks that are made in ITERATIVEALG are checking whether or not two trees are isomorphic (in the phylogenetic sense), and checking whether or not two trees are isomorphic is polynomial-time. In fact, the authors of [29] showed checking whether or not two trees are isomorphic is linear-time, and the author of [30] showed that checking whether or not two trees are isomorphic is alogtime. Now, we will construct a one to one function from iterations of ITERATIVEALG to elements in $(X_{Z_1})^L \cup [(X_{Z_1})^L]^*$, where $[(X_{Z_1})^L]^* = \{x^* : x \in (X_{Z_1})^L\}$. Since the maximum number of steps between iterations is a constant, the above construction will show that ITERATIVEALG runs in polynomial-time. First we will construct a one to one function

from iterations of ITERATIVEALG to $\{1, 3, \dots, 2p-3, 2p-1\}$, where $p \in \mathbb{N}$, given that ITERATIVEALG stops at p . There are two ways in which ITERATIVEALG can be iterated. The first way is in Step 9 when we are sent back to Step 6, and we replace i with $i+2$ whilst not changing j . The second way is in Step 10 when we are sent back to Step 3, and we replace j with $j+1$ and i with $m+1$, where $m+1 \in \{i+2, i+4, i+6, \dots\}$. Since at the start of ITERATIVEALG $i = 1$, we can construct a one to one function from iterations of ITERATIVEALG to $\{1, 3, \dots, 2p-3, 2p-1\}$. Now we construct a one to one function from $\{1, 3, \dots, 2p-3, 2p-1\}$ to $(X_{Z_1})^L \cup [(X_{Z_1})^L]^*$. Let n be any element in $\{1, 3, \dots, 2p-3, 2p-1\}$ then, by Lemma 5.21 and Lemma 5.22, $(X_{S_{D_n}})^L \neq \emptyset$. If $n = 2p-1$ then send n to any element in $(X_{S_{D_n}})^L$. If $n \neq 2p-1$ then $n \in \{1, 3, \dots, 2p-3\}$, and, by Lemma 5.21 and Lemma 5.22, $(X_{S_{D_{n+2}}})^L \neq \emptyset, \dots, (X_{S_{D_{2p-1}}})^L \neq \emptyset$. If $(X_{S_{D_n}})^L \setminus ((X_{S_{D_{n+2}}})^L \cup \dots \cup (X_{S_{D_{2p-1}}})^L) \neq \emptyset$ then send n to any element in $(X_{S_{D_n}})^L \setminus ((X_{S_{D_{n+2}}})^L \cup \dots \cup (X_{S_{D_{2p-1}}})^L)$. If $(X_{S_{D_n}})^L \setminus ((X_{S_{D_{n+2}}})^L \cup \dots \cup (X_{S_{D_{2p-1}}})^L) = \emptyset$ then $(X_{S_{D_n}})^L = (X_{Z_j})^L$ for some $j \in \mathbb{N}$, because if $(X_{S_{D_n}})^L \neq (X_{Z_j})^L$ for all $j \in \mathbb{N}$ then $(X_{S_{D_n}})^L \setminus ((X_{S_{D_{n+2}}})^L \cup \dots \cup (X_{S_{D_{2p-1}}})^L) = (X_{S_{D_n}})^L$, and since $(X_{S_{D_n}})^L \neq \emptyset$, we have $(X_{S_{D_n}})^L \setminus ((X_{S_{D_{n+2}}})^L \cup \dots \cup (X_{S_{D_{2p-1}}})^L) \neq \emptyset$. Consider $(X_{S_{D_{n+2}}})^L \setminus ((X_{S_{D_{n+4}}})^L \cup \dots \cup (X_{S_{D_{2p-1}}})^L)$. If $(X_{S_{D_{n+2}}})^L \setminus ((X_{S_{D_{n+4}}})^L \cup \dots \cup (X_{S_{D_{2p-1}}})^L) = \emptyset$ then $(X_{S_{D_{n+2}}})^L = (X_{Z_{j+1}})^L$, but, by Step 7, $(X_{Z_{j+1}})^L \subsetneq (X_{Z_j})^L$ hence $(X_{S_{D_n}})^L \setminus ((X_{S_{D_{n+2}}})^L \cup \dots \cup (X_{S_{D_{2p-1}}})^L) \neq \emptyset$ - a contradiction. Therefore $(X_{S_{D_{n+2}}})^L \setminus ((X_{S_{D_{n+4}}})^L \cup \dots \cup (X_{S_{D_{2p-1}}})^L) \neq \emptyset$ thus $[(X_{S_{D_{n+2}}})^L]^* \setminus ([(X_{S_{D_{n+4}}})^L]^* \cup \dots \cup [(X_{S_{D_{2p-1}}})^L]^*) \neq \emptyset$, so send n to any element in $[(X_{S_{D_{n+2}}})^L]^* \setminus ([(X_{S_{D_{n+4}}})^L]^* \cup \dots \cup [(X_{S_{D_{2p-1}}})^L]^*)$. Now we will show that this function, call it g , is one to one. Take any $\alpha, \beta \in \{1, 3, \dots, 2p-3, 2p-1\}$ such that $\alpha \neq \beta$. We see that α is either sent to an element in $(X_{S_{D_\alpha}})^L \setminus ((X_{S_{D_{\alpha+2}}})^L \cup \dots \cup (X_{S_{D_{2p-1}}})^L)$ or to an element in $[(X_{S_{D_{\alpha+2}}})^L]^* \setminus ([(X_{S_{D_{\alpha+4}}})^L]^* \cup \dots \cup [(X_{S_{D_{2p-1}}})^L]^*)$. Similarly with β , so if α is sent to an element in $(X_{S_{D_\alpha}})^L \setminus ((X_{S_{D_{\alpha+2}}})^L \cup \dots \cup (X_{S_{D_{2p-1}}})^L)$ whilst β is sent to an element in $[(X_{S_{D_{\beta+2}}})^L]^* \setminus ([(X_{S_{D_{\beta+4}}})^L]^* \cup \dots \cup [(X_{S_{D_{2p-1}}})^L]^*)$, or vice versa, then $g(\alpha) \neq g(\beta)$. Without loss of generality, suppose that $\alpha < \beta$, and suppose that α is sent to an element in $(X_{S_{D_\alpha}})^L \setminus ((X_{S_{D_{\alpha+2}}})^L \cup \dots \cup (X_{S_{D_{2p-1}}})^L)$ whilst β is sent to an element in $(X_{S_{D_\beta}})^L \setminus ((X_{S_{D_{\beta+2}}})^L \cup \dots \cup (X_{S_{D_{2p-1}}})^L)$. Since $\alpha < \beta$, we see that $[(X_{S_{D_\alpha}})^L \setminus ((X_{S_{D_{\alpha+2}}})^L \cup \dots \cup (X_{S_{D_{2p-1}}})^L)] \cap (X_{S_{D_\beta}})^L = \emptyset$, so that $g(\alpha) \neq g(\beta)$. Similarly with the other case. Thus g is a one to one function; hence there is a one to one function from

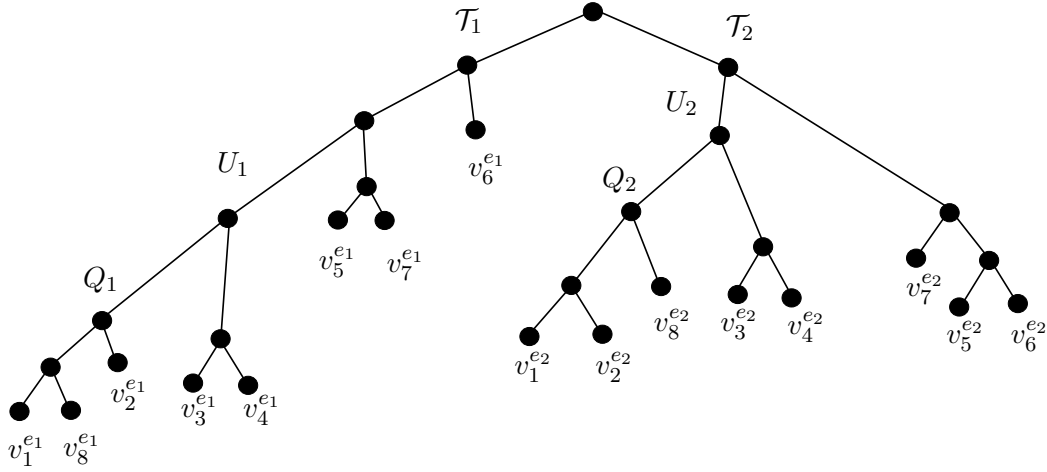
iterations of ITERATIVEALG to $(X_{Z_1})^L \cup [(X_{Z_1})^L]^*$. There are a maximum of 8 steps between iterations, whilst there are two steps at the beginning of ITERATIVEALG that only feature in the first iteration. Therefore the maximum number of steps that ITERATIVEALG can perform is less than or equal to $16|(X_{Z_1})^L| + 2$. \square

5.4 A Sufficient Condition

In this section a sufficient condition for a network to display a tree twice is presented. The sufficient condition relies on some new concepts and observations. The first new concept is that of a chain of tree pairs, which is a series of tree pairs such that the tree pair following another tree pair in the series either has its first element as a maximal subtree of the first element of the tree pair that proceeds and likewise with the second element, or the tree pair that follows another tree pair is the result of the proceeding tree pair being inputted into ITERATIVEALG. The second new concept is that of a major tree pair that is not a sterile major tree pair, which we call a fertile major tree pair, and these major tree pairs have the property that one of the maximal subtrees of the first element makes a major tree pair with one of the maximal subtrees of the second element. The third new concept is that of a sterile major tree pair that meets the input criteria of ITERATIVEALG and yields a major tree pair together with sets upon input into ITERATIVEALG, which we call a productive sterile major tree pair. This leads to the concept of a chain of tree pairs where every link in the chain is either a fertile major tree pair or a productive sterile major tree pair, which we call a chain of fertile/productive sterile major tree pairs. A simple observation would be that a chain of fertile/productive sterile major tree pairs must eventually end at either a trivial tree pair or a non-productive sterile major tree pair. A chain of tree pairs where every link except the last is a fertile major tree pair or a productive sterile major tree pair is called a maximal chain of fertile/productive sterile major tree pairs.

Definition 5.11. A *chain of tree pairs* is a series of tree pairs $(U_1^1, U_2^1), (U_1^2, U_2^2), \dots, (U_1^n, U_2^n)$, where $n \in \mathbb{N}$, such that either U_1^j is a maximal subtree of U_1^{j-1} and U_2^j is a maximal subtree of U_2^{j-1} or (U_1^j, U_2^j) is the output of (U_1^{j-1}, U_2^{j-1}) being inputted into ITERATIVEALG, for all $j \in \{2, \dots, n\}$.

Example 5.10. An example of a chain of tree pairs in the following figure is $(\mathcal{T}_1, \mathcal{T}_2), (U_1, U_2), (Q_1, Q_2)$, as (U_1, U_2) is the result of $(\mathcal{T}_1, \mathcal{T}_2)$ being inputted into ITERATIVEALG, whilst Q_1 is a maximal subtree of U_1 and Q_2 is a maximal subtree of U_2 .



Definition 5.12. A major tree pair (U_1, U_2) of a representation \mathcal{T} is a *fertile major tree pair* of \mathcal{T} when one of the maximal subtrees of U_1 makes a major tree pair with one of the maximal subtrees of U_2 .

Example 5.11. The major tree pair (U_1, U_2) from Example 5.10 is a fertile major tree pair, as (Q_1, Q_2) is a major tree pair.

Definition 5.13. A sterile major tree pair (U_1, U_2) of a representation \mathcal{T} is called a *productive sterile major tree pair* when (U_1, U_2) meets the input criteria of ITERATIVEALG, and (U_1, U_2) yields a major tree pair together with sets when inputted into ITERATIVEALG.

Example 5.12. The sterile major tree pair $(\mathcal{T}_1, \mathcal{T}_2)$ from Example 5.10 is a productive sterile major tree pair.

Definition 5.14. A chain of tree pairs is a *chain of fertile/(productive sterile) major tree pairs* when every link in the chain is a fertile major tree pair or a productive sterile major tree pair.

Example 5.13. The chain of tree pairs $(\mathcal{T}_1, \mathcal{T}_2), (U_1, U_2)$ from Example 5.10 is a chain of fertile/(productive sterile) major tree pairs.

Definition 5.15. A chain of tree pairs $(U_1^1, U_2^1), (U_1^2, U_2^2), \dots, (U_1^n, U_2^n)$, where $n \in \mathbb{N}$, is a *maximal chain of fertile/(productive sterile) major tree pairs* when $(U_1^1, U_2^1), \dots, (U_1^{n-1}, U_2^{n-1})$ is a chain of fertile/(productive sterile) major tree pairs and (U_1^n, U_2^n) is either a non-productive sterile major tree pair or a trivial tree pair.

Note 8. In the above definition we allow a maximal chain of fertile/(productive sterile) major tree pairs that consists of one tree pair.

Example 5.14. The chain of tree pairs $(\mathcal{T}_1, \mathcal{T}_2), (U_1, U_2), (Q_1, Q_2)$ from Example 5.10 is a maximal chain of fertile/(productive sterile) major tree pairs, as (Q_1, Q_2) is a non-productive sterile major tree pair.

Lemma 5.23. *Let (U_1, U_2) be a major tree pair of \mathcal{T} . Let A_{U_1}, M_{U_1} be the two maximal subtrees of U_1 , and let A_{U_2}, M_{U_2} be the two maximal subtrees of U_2 . If (U_1, U_2) is a fertile major tree pair then either (A_{U_1}, A_{U_2}) and (M_{U_1}, M_{U_2}) are tree pairs or (A_{U_1}, M_{U_2}) and (M_{U_1}, A_{U_2}) are tree pairs.*

Proof. By the definition of a fertile major tree pair, one of $(A_{U_1}, A_{U_2}), (M_{U_1}, M_{U_2}), (A_{U_1}, M_{U_2})$, or (M_{U_1}, A_{U_2}) is a major tree pair. Without loss of generality, suppose that (A_{U_1}, A_{U_2}) is a major tree pair then $(X_{A_{U_1}})^L = (X_{A_{U_2}})^L$. Now $(X_{A_{U_1}})^L \cap (X_{M_{U_1}})^L = \emptyset$ and $(X_{A_{U_1}})^L \cup (X_{M_{U_1}})^L = (X_{U_1})^L$, so $(X_{M_{U_1}})^L = (X_{U_1})^L \setminus (X_{A_{U_1}})^L$. Similarly for $(X_{A_{U_2}})^L$ and $(X_{M_{U_2}})^L$, so $(X_{M_{U_2}})^L = (X_{U_2})^L \setminus (X_{A_{U_2}})^L$. Since $(X_{U_1})^L = (X_{U_2})^L$ and $(X_{A_{U_1}})^L = (X_{A_{U_2}})^L$, we have $(X_{M_{U_1}})^L = (X_{U_1})^L \setminus (X_{A_{U_1}})^L = (X_{U_2})^L \setminus (X_{A_{U_2}})^L = (X_{M_{U_2}})^L$. Therefore (M_{U_1}, M_{U_2}) is a tree pair. \square

Lemma 5.24. *A chain of fertile/(productive sterile) major tree pairs can be extended to a maximal chain of fertile/(productive sterile) major tree pairs.*

Proof. Since we can not have a never ending chain of fertile/(productive sterile) major tree pairs, we must eventually get a maximal chain of fertile/(productive sterile) major tree pairs. \square

Proposition 5.10. *Let \mathcal{T} be a basic⁰ representation of a basic cyclic pair with maximal subtrees $\mathcal{T}_1, \mathcal{T}_2$ such that $(\mathcal{T}_1)^L \neq (\mathcal{T}_2)^L$, and $C_{\mathcal{T}} = \emptyset$. If every sterile major tree pair (Q_1, Q_2) that is the last tree pair in a maximal chain of fertile/(productive sterile) major tree pairs starting at $(\mathcal{T}_1, \mathcal{T}_2)$, where Q_1 has maximal subtrees A_{Q_1}, M_{Q_1} and Q_2 has maximal subtrees A_{Q_2}, M_{Q_2} , has $(M_{Q_1})^L \mid \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) = (M_{Q_2})^L \mid \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$, $(A_{Q_1})^L \mid \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) = (A_{Q_2})^L \mid \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right)$, $(M_{Q_2})^L \mid \left((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \right) = (A_{Q_1})^L \mid \left((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \right)$, and $(A_{Q_2})^L \mid \left((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \right) = (M_{Q_1})^L \mid \left((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \right)$ or ITERATIVEALG returns the statement “displays a tree twice” upon input of (Q_1, Q_2) . Then \mathcal{T} displays a tree twice.*

Proof. We will construct two sets R and W that together show that \mathcal{T} displays a tree twice. Let $R_{\mathcal{T}_1}$ be a subset of $X_{\mathcal{T}_1}$, and let $R_{\mathcal{T}_2}$ be a subset of $X_{\mathcal{T}_2}$. Let (U_1, U_2) be any tree pair of \mathcal{T} . Let the two maximal subtrees of U_1 be A_{U_1}, M_{U_1} , and let the two maximal subtrees of U_2 be A_{U_2}, M_{U_2} . If (U_1, U_2) is a sterile major tree pair then (U_1, U_2)

can be either a productive or non-productive sterile major tree pair. If (U_1, U_2) is a non-productive sterile major tree pair then (A_{U_1}, A_{U_2}) , (M_{U_1}, M_{U_2}) , (A_{U_1}, M_{U_2}) , (M_{U_1}, A_{U_2}) are all not tree pairs. If $(M_{U_1})^L \mid \left((X_{M_{U_1}})^L \cap (X_{M_{U_2}})^L \right) = (M_{U_2})^L \mid \left((X_{M_{U_1}})^L \cap (X_{M_{U_2}})^L \right)$, $(A_{U_1})^L \mid \left((X_{A_{U_1}})^L \cap (X_{A_{U_2}})^L \right) = (A_{U_2})^L \mid \left((X_{A_{U_1}})^L \cap (X_{A_{U_2}})^L \right)$, $(M_{U_2})^L \mid \left((X_{M_{U_2}})^L \cap (X_{A_{U_1}})^L \right) = (A_{U_1})^L \mid \left((X_{M_{U_2}})^L \cap (X_{A_{U_1}})^L \right)$, and $(A_{U_2})^L \mid \left((X_{A_{U_2}})^L \cap (X_{M_{U_1}})^L \right) = (M_{U_1})^L \mid \left((X_{A_{U_2}})^L \cap (X_{M_{U_1}})^L \right)$ then define $(R_{\mathcal{T}_1} \cap X_{A_{U_1}})$ as having the property that $(R_{\mathcal{T}_1} \cap X_{A_{U_1}})^L = (X_{A_{U_1}})^L \cap (X_{A_{U_2}})^L$, and define $(R_{\mathcal{T}_1} \cap X_{M_{U_1}})$ as having the property that $(R_{\mathcal{T}_1} \cap X_{M_{U_1}})^L = (X_{M_{U_1}})^L \cap (X_{M_{U_2}})^L$. Similarly, define $(R_{\mathcal{T}_2} \cap X_{A_{U_2}})$ as having the property that $(R_{\mathcal{T}_2} \cap X_{A_{U_2}})^L = (X_{A_{U_2}})^L \cap (X_{M_{U_1}})^L$, and define $(R_{\mathcal{T}_2} \cap X_{M_{U_2}})$ as having the property that $(R_{\mathcal{T}_2} \cap X_{M_{U_2}})^L = (X_{M_{U_2}})^L \cap (X_{A_{U_1}})^L$. If **ITERATIVEALG** returns the statement “displays a tree twice” upon input of (U_1, U_2) then we have $(U_1)^L \mid (R_{U_{f(j+1)}}^q)^L = (U_2)^L \mid (R_{U_{f(j+1)}}^q)^L$ and $(U_1)^L \mid (R_{U_{f(j)}}^q)^L = (U_2)^L \mid (R_{U_{f(j)}}^q)^L$ for some $j, q \in \mathbb{N}$, where $(R_{U_{f(j)}}^q)^L = (X_{U_{f(j)}})^L \setminus (R_{U_{f(j+1)}}^q)^L$, $(R_{U_{f(j)}}^q)^L \subseteq (X_{U_{f(j)}})^L$, and $(R_{U_{f(j+1)}}^q)^L \subseteq (X_{U_{f(j+1)}})^L$. Define $(R_{\mathcal{T}_1} \cap (X_{U_1}))$ as having the property that $(R_{\mathcal{T}_1} \cap X_{U_1})^L = (R_{U_1}^q)^L$, and define $(R_{\mathcal{T}_2} \cap (X_{U_2}))$ as having the property that $(R_{\mathcal{T}_2} \cap X_{U_2})^L = (R_{U_2}^q)^L$. If (U_1, U_2) is a productive sterile major tree pair then (U_1, U_2) yields a major tree pair (Y_1, Y_2) together with sets $(R_{U_1}^{(j)*})^L$, $(R_{U_2}^{(j)*})^L$, and $(X_{Z_j})^L$, for some $j \in \mathbb{N}$, when (U_1, U_2) is inputted into **ITERATIVEALG**. As in Proposition 5.7, define $(R_{\mathcal{T}_1} \cap X_{U_1})$ as having the property that $(R_{\mathcal{T}_1} \cap X_{U_1})^L = (R_{U_1}^{(j)*})^L \cup ((R_{\mathcal{T}_1})^L \cap (X_{Y_1})^L)$. Similarly, define $(R_{\mathcal{T}_2} \cap X_{U_2})$ as having the property that $(R_{\mathcal{T}_2} \cap X_{U_2})^L = (R_{U_2}^{(j)*})^L \cup ((R_{\mathcal{T}_2})^L \cap (X_{Y_2})^L)$. If (U_1, U_2) is a trivial tree pair then $(U_1)^L = (U_2)^L$. Define $R_{\mathcal{T}_1} \cap X_{U_1}$ as X_{U_1} , and define $R_{\mathcal{T}_2} \cap X_{U_2}$ as the empty set. Let R be $R_{\mathcal{T}_1} \cup R_{\mathcal{T}_2}$. We claim that R is a switching of \mathcal{T} . In order to prove that claim, we need to show that for each $v_i^{e_m} \in X_{\mathcal{T}}$, where $m \in \{1, 2\}$ and $i \in \mathbb{N}$, exactly one of $v_i^{e_m}$, $v_i^{e_n}$, where $n \in \{1, 2\} \setminus \{m\}$, is in R . Take any $v_i^{e_m} \in X_{\mathcal{T}}$ then, since \mathcal{T} is a basic^0 representation, either $v_i^{e_m} \in X_{\mathcal{T}_1}$ and $v_i^{e_n} \in X_{\mathcal{T}_2}$ or $v_i^{e_m} \in X_{\mathcal{T}_2}$ and $v_i^{e_n} \in X_{\mathcal{T}_1}$. Without loss of generality, suppose that $v_i^{e_m} \in X_{\mathcal{T}_1}$ and $v_i^{e_n} \in X_{\mathcal{T}_2}$. Since $(\mathcal{T}_1)^L \neq (\mathcal{T}_2)^L$ and \mathcal{T} is a basic^0 representation, $(\mathcal{T}_1, \mathcal{T}_2)$ is a major tree pair of \mathcal{T} , so $(\mathcal{T}_1, \mathcal{T}_2)$ is either a sterile major tree pair or a fertile major tree pair. Suppose that $(\mathcal{T}_1, \mathcal{T}_2)$ is a sterile major tree pair; let $A_{\mathcal{T}_1}, M_{\mathcal{T}_1}$ be the two maximal subtrees of \mathcal{T}_1 , and let $A_{\mathcal{T}_2}, M_{\mathcal{T}_2}$ be the two maximal subtrees of \mathcal{T}_2 . If $(M_{\mathcal{T}_1})^L \mid \left((X_{M_{\mathcal{T}_1}})^L \cap (X_{M_{\mathcal{T}_2}})^L \right) = (M_{\mathcal{T}_2})^L \mid \left((X_{M_{\mathcal{T}_1}})^L \cap (X_{M_{\mathcal{T}_2}})^L \right)$, $(A_{\mathcal{T}_1})^L \mid \left((X_{A_{\mathcal{T}_1}})^L \cap (X_{A_{\mathcal{T}_2}})^L \right) = (A_{\mathcal{T}_2})^L \mid \left((X_{A_{\mathcal{T}_1}})^L \cap (X_{A_{\mathcal{T}_2}})^L \right)$, $(M_{\mathcal{T}_2})^L \mid \left((X_{M_{\mathcal{T}_2}})^L \cap (X_{A_{\mathcal{T}_1}})^L \right) = (A_{\mathcal{T}_1})^L \mid \left((X_{M_{\mathcal{T}_2}})^L \cap (X_{A_{\mathcal{T}_1}})^L \right)$, and

$(A_{\mathcal{T}_2})^L \mid \left((X_{A_{\mathcal{T}_2}})^L \cap (X_{M_{\mathcal{T}_1}})^L \right) = (M_{\mathcal{T}_1})^L \mid \left((X_{A_{\mathcal{T}_2}})^L \cap (X_{M_{\mathcal{T}_1}})^L \right)$ then, since $v_i^{em} \in X_{\mathcal{T}_1}$ and $v_i^{en} \in X_{\mathcal{T}_2}$, either $v_i^{em} \in X_{A_{\mathcal{T}_1}}$ or $v_i^{em} \in X_{M_{\mathcal{T}_1}}$ and either $v_i^{en} \in X_{A_{\mathcal{T}_2}}$ or $v_i^{en} \in X_{M_{\mathcal{T}_2}}$. Suppose that $v_i^{em} \in X_{A_{\mathcal{T}_1}}$ and $v_i^{en} \in X_{M_{\mathcal{T}_2}}$ then $v_i^{em} \notin X_{M_{\mathcal{T}_1}}$ and $v_i^{en} \notin X_{A_{\mathcal{T}_2}}$. Thus $\ell_{v_i} \in (X_{A_{\mathcal{T}_1}})^L \cap (X_{M_{\mathcal{T}_2}})^L$, $\ell_{v_i} \notin (X_{M_{\mathcal{T}_1}})^L \cap (X_{M_{\mathcal{T}_2}})^L$, and $\ell_{v_i} \notin (X_{A_{\mathcal{T}_1}})^L \cap (X_{A_{\mathcal{T}_2}})^L$. Hence, by definition, $\ell_{v_i} \in (R_{\mathcal{T}_2} \cap X_{M_{\mathcal{T}_2}})^L$, so that $v_i^{en} \in R_{\mathcal{T}_2}$. We also have $\ell_{v_i} \notin (R_{\mathcal{T}_1} \cap X_{A_{\mathcal{T}_1}})^L$ and $\ell_{v_i} \notin (R_{\mathcal{T}_1} \cap X_{M_{\mathcal{T}_1}})^L$, so $v_i^{em} \notin R_{\mathcal{T}_1}$. Of course, $v_i^{em} \notin R_{\mathcal{T}_2}$ and $v_i^{en} \notin R_{\mathcal{T}_1}$, because $v_i^{em} \notin X_{\mathcal{T}_2}$ and $v_i^{en} \notin X_{\mathcal{T}_1}$. Therefore $v_i^{en} \in R$ and $v_i^{em} \notin R$. The other cases are similar. If ITERATIVEALG returns the statement “displays a tree twice” upon input of $(\mathcal{T}_1, \mathcal{T}_2)$ then $\ell_{v_i} \in (R_{U_1}^q)^L$ or $\ell_{v_i} \in (R_{U_2}^q)^L$. Without loss of generality, suppose that $\ell_{v_i} \in (R_{U_1}^q)^L$ then $\ell_{v_i} \notin (R_{U_2}^q)^L$, by construction. Hence $\ell_{v_i} \in (R_{\mathcal{T}_1})^L$, so that, since $v_i^{em} \in X_{\mathcal{T}_1}$, we have $v_i^{em} \in R_{\mathcal{T}_1}$. We see that $v_i^{en} \notin R_{\mathcal{T}_2}$, because $\ell_{v_i} \notin (R_{U_2}^q)^L$. We also see that $v_i^{en} \notin R_{\mathcal{T}_1}$, because $v_i^{en} \in X_{\mathcal{T}_2}$ and $R_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$. Therefore $v_i^{em} \in R$ and $v_i^{en} \notin R$.

Suppose that $(\mathcal{T}_1, \mathcal{T}_2)$ is a fertile major tree pair then, by Lemma 5.23, both of the maximal subtrees of \mathcal{T}_1 make tree pairs with the two maximal subtrees of \mathcal{T}_2 . Without loss of generality, suppose that $(A_{\mathcal{T}_1}, A_{\mathcal{T}_2})$ and $(M_{\mathcal{T}_1}, M_{\mathcal{T}_2})$ are tree pairs; then we can have either $v_i^{em} \in X_{A_{\mathcal{T}_1}}$ and $v_i^{en} \in X_{A_{\mathcal{T}_2}}$ or $v_i^{em} \in X_{M_{\mathcal{T}_1}}$ and $v_i^{en} \in X_{M_{\mathcal{T}_2}}$. Without loss of generality, suppose that $v_i^{em} \in X_{A_{\mathcal{T}_1}}$ and $v_i^{en} \in X_{A_{\mathcal{T}_2}}$. Now $(A_{\mathcal{T}_1}, A_{\mathcal{T}_2})$ can be either a trivial tree pair or a major tree pair. If $(A_{\mathcal{T}_1}, A_{\mathcal{T}_2})$ is a major tree pair then it can either a fertile major tree pair or a sterile major tree pair. If $(A_{\mathcal{T}_1}, A_{\mathcal{T}_2})$ is a non-productive sterile major tree pair then we can show that exactly one of v_i^{em}, v_i^{en} is in R .

Suppose that $(A_{\mathcal{T}_1}, A_{\mathcal{T}_2})$ is a trivial tree pair. Then $R_{\mathcal{T}_1} \cap X_{A_{\mathcal{T}_1}}$ is defined as $X_{A_{\mathcal{T}_1}}$, and $R_{\mathcal{T}_2} \cap X_{A_{\mathcal{T}_2}}$ is defined as the empty set. Thus $v_i^{em} \in R_{\mathcal{T}_1}$ and $v_i^{en} \notin R_{\mathcal{T}_2}$, so that $v_i^{em} \in R$ and $v_i^{en} \notin R$. Suppose that $(\mathcal{T}_1, \mathcal{T}_2)$ is a productive sterile major tree pair then $(\mathcal{T}_1, \mathcal{T}_2)$ yields a major tree pair (Y_1, Y_2) together with sets $(R_{\mathcal{T}_1}^{(j)*})^L$, $(R_{\mathcal{T}_2}^{(j)*})^L$, and $(X_{Z_j})^L$, for some $j \in \mathbb{N}$, when $(\mathcal{T}_1, \mathcal{T}_2)$ is inputted into ITERATIVEALG. By Lemma 5.14, $(X_{\mathcal{T}_1})^L = (X_{\mathcal{T}_2})^L = (R_{\mathcal{T}_1}^{(j)*})^L \cup (R_{\mathcal{T}_2}^{(j)*})^L \cup (X_{Z_j})^L$, $(R_{\mathcal{T}_1}^{(j)*})^L \cap (R_{\mathcal{T}_2}^{(j)*})^L = \emptyset$, $(R_{\mathcal{T}_1}^{(j)*})^L \cap (X_{Z_j})^L = \emptyset$, and $(R_{\mathcal{T}_2}^{(j)*})^L \cap (X_{Z_j})^L = \emptyset$. If $\ell_{v_i} \in (R_{\mathcal{T}_1}^{(j)*})^L$ then $\ell_{v_i} \notin (R_{\mathcal{T}_2}^{(j)*})^L$ and $\ell_{v_i} \notin (X_{Z_j})^L$; hence $\ell_{v_i} \in (R_{\mathcal{T}_1} \cap X_{\mathcal{T}_1})^L$ and $\ell_{v_i} \notin (R_{\mathcal{T}_2} \cap X_{\mathcal{T}_2})^L$, because $(X_{Y_1})^L = (X_{Y_2})^L = (X_{Z_j})^L$. Therefore $v_i^{em} \in R_{\mathcal{T}_1} \cap X_{\mathcal{T}_1}$ and $v_i^{en} \notin R_{\mathcal{T}_2} \cap X_{\mathcal{T}_2}$; thus $v_i^{em} \in R$ and $v_i^{en} \notin R$. Similarly, if $\ell_{v_i} \in (R_{\mathcal{T}_2}^{(j)*})^L$ then $v_i^{en} \in R$ and $v_i^{em} \notin R$. If $\ell_{v_i} \in (X_{Z_j})^L$ then $\ell_{v_i} \in (X_{Y_1})^L = (X_{Y_2})^L$. We must have $v_i^{em} \in (X_{Y_1})$ and $v_i^{en} \in (X_{Y_2})$. If (Y_1, Y_2) is a non-productive sterile major tree pair then we can show that either $v_i^{en} \in R$ and $v_i^{em} \notin R$ or $v_i^{em} \in R$ and $v_i^{en} \notin R$. If $(A_{\mathcal{T}_1}, A_{\mathcal{T}_2})$ and (Y_1, Y_2) are fertile major tree pairs then we can apply the same argument to $(A_{\mathcal{T}_1}, A_{\mathcal{T}_2})$ and (Y_1, Y_2) as we did to $(\mathcal{T}_1, \mathcal{T}_2)$. Likewise, if $(A_{\mathcal{T}_1}, A_{\mathcal{T}_2})$ and (Y_1, Y_2) are productive sterile major tree pairs. In this way and by Lemma 5.24, we can get a maximal chain of fertile/(productive sterile)

major tree pairs $(U_1^1, U_2^1), (U_1^2, U_2^2), \dots, (U_1^n, U_2^n)$ where $(U_1^1, U_2^1) = (\mathcal{T}_1, \mathcal{T}_2)$, $(U_1^2, U_2^2) = (A_{\mathcal{T}_1}, A_{\mathcal{T}_2})$ or $(U_1^2, U_2^2) = (Y_1, Y_2)$, and $v_i^{e^m} \in X_{U_1^j}$ and $v_i^{e^n} \in X_{U_2^j}$ for all $j \in (1, 2, \dots, n)$. Therefore $v_i^{e^m} \in X_{U_1^n}$ and $v_i^{e^n} \in X_{U_2^n}$ where (U_1^n, U_2^n) is either a trivial tree pair or a sterile major tree pair. In both cases we see that R is a switching of \mathcal{T} .

Next we define another two sets $W_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$ and $W_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$ where $W_{\mathcal{T}_1}$ has the property that $(W_{\mathcal{T}_1})^L = (R_{\mathcal{T}_2})^L$ and $W_{\mathcal{T}_2}$ has the property that $(W_{\mathcal{T}_2})^L = (R_{\mathcal{T}_1})^L$. We define W as $W_{\mathcal{T}_1} \cup W_{\mathcal{T}_2}$. Since R is a switching of \mathcal{T} , W is a switching of \mathcal{T} . Now we will show that either $v^{e_1} \in R$ and $v^{e_2} \in W$ or $v^{e_2} \in R$ and $v^{e_1} \in W$. Since R is a switching of \mathcal{T} , either $v^{e_1} \in R$ or $v^{e_2} \in R$. Without loss of generality, suppose that $v^{e_1} \in R$. Either $v^{e_1} \in X_{\mathcal{T}_1}$ or $v^{e_1} \in X_{\mathcal{T}_2}$. Without loss of generality, suppose that $v^{e_1} \in X_{\mathcal{T}_1}$ then $v^{e_1} \in R_{\mathcal{T}_1}$ and $\ell_v \in (R_{\mathcal{T}_1})^L$. Hence $\ell_v \in (W_{\mathcal{T}_2})^L$. Since $W_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$ and $v^{e_2} \in X_{\mathcal{T}_2}$, we have $v^{e_2} \in W_{\mathcal{T}_2}$. Thus $v^{e_2} \in W$.

The final task is to show that $(\mathcal{T}|R)^L = (\mathcal{T}|W)^L$. First we will show that if (U_1, U_2) is a tree pair of \mathcal{T} then $(U_1|R_{\mathcal{T}_1} \cap X_{U_1})^L = (U_2|W_{\mathcal{T}_2} \cap X_{U_2})^L$ and $(U_2|R_{\mathcal{T}_2} \cap X_{U_2})^L = (U_1|W_{\mathcal{T}_1} \cap X_{U_1})^L$. Either (U_1, U_2) is a trivial or a major tree pair. Suppose that (U_1, U_2) is a trivial tree pair then $(U_1)^L = (U_2)^L$, $R_{\mathcal{T}_1} \cap X_{U_1} = X_{U_1}$, and $R_{\mathcal{T}_2} \cap X_{U_2} = \emptyset$. Now $(R_{\mathcal{T}_1} \cap X_{U_1})^L = (X_{U_1})^L$, and it is easy to show that $(R_{\mathcal{T}_1} \cap X_{U_1})^L = (R_{\mathcal{T}_1})^L \cap (X_{U_1})^L$. It is also easy to show that $(U_1|R_{\mathcal{T}_1} \cap X_{U_1})^L = (U_1)^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L$. Since $(U_1)^L = (U_2)^L$, we just need to show that $(R_{\mathcal{T}_1} \cap X_{U_1})^L = (W_{\mathcal{T}_2} \cap X_{U_2})^L$. Since $(X_{U_1})^L = (X_{U_2})^L$ and $(R_{\mathcal{T}_1})^L = (W_{\mathcal{T}_2})^L$, we have $(R_{\mathcal{T}_1})^L \cap (X_{U_1})^L = (W_{\mathcal{T}_2})^L \cap (X_{U_2})^L$. Hence $(R_{\mathcal{T}_1} \cap X_{U_1})^L = (W_{\mathcal{T}_2} \cap X_{U_2})^L$. Therefore $(U_1)^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L = (U_2)^L | (W_{\mathcal{T}_2} \cap X_{U_2})^L$. It follows that $(U_1|R_{\mathcal{T}_1} \cap X_{U_1})^L = (U_2|W_{\mathcal{T}_2} \cap X_{U_2})^L$. Similarly, we can show that $(U_2|R_{\mathcal{T}_2} \cap X_{U_2})^L = (U_1|W_{\mathcal{T}_1} \cap X_{U_1})^L$ (note that we get the result without using our definitions of $R_{\mathcal{T}_1} \cap X_{U_1}$ and $R_{\mathcal{T}_2} \cap X_{U_2}$).

Suppose that (U_1, U_2) is a major tree pair then it is either a sterile major tree pair or a fertile major tree pair. Suppose that (U_1, U_2) is a sterile major tree pair then (U_1, U_2) is either a productive sterile major tree pair or a non-productive sterile major tree pair. Suppose that (U_1, U_2) is a non-productive sterile major tree pair. Let the two maximal subtrees of U_1 be A_{U_1} , M_{U_1} , and let the two maximal subtrees of U_2 be A_{U_2} , M_{U_2} . Suppose that $(M_{U_1})^L | \left((X_{M_{U_1}})^L \cap (X_{M_{U_2}})^L \right) = (M_{U_2})^L | \left((X_{M_{U_1}})^L \cap (X_{M_{U_2}})^L \right)$, $(A_{U_1})^L | \left((X_{A_{U_1}})^L \cap (X_{A_{U_2}})^L \right) = (A_{U_2})^L | \left((X_{A_{U_1}})^L \cap (X_{A_{U_2}})^L \right)$, $(M_{U_2})^L | \left((X_{M_{U_2}})^L \cap (X_{A_{U_1}})^L \right) = (A_{U_1})^L | \left((X_{M_{U_2}})^L \cap (X_{A_{U_1}})^L \right)$, and $(A_{U_2})^L | \left((X_{A_{U_2}})^L \cap (X_{M_{U_1}})^L \right) = (M_{U_1})^L | \left((X_{A_{U_2}})^L \cap (X_{M_{U_1}})^L \right)$ then first we will show that $(U_1|R_{\mathcal{T}_1} \cap X_{U_1})^L = (U_2|W_{\mathcal{T}_2} \cap X_{U_2})^L$. Note that $(R_{\mathcal{T}_1} \cap X_{U_1})^L = (W_{\mathcal{T}_2} \cap X_{U_2})^L$, $(U_1|R_{\mathcal{T}_1} \cap X_{U_1})^L = (U_1)^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L$, and $(U_2|W_{\mathcal{T}_2} \cap X_{U_2})^L = (U_2)^L | (W_{\mathcal{T}_2} \cap X_{U_2})^L$, so if

we can show that $(U_1)^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L = (U_2)^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L$ then that will suffice. In order to show that we need to show that one maximal subtree of $(U_1)^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L$ is equal to a maximal subtree of $(U_2)^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L$ whilst the remaining pair of maximal subtrees are also equal. We see that the maximal subtrees of $(U_1)^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L$ are $(A_{U_1})^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L \cap (X_{A_{U_1}})^L$ and $(M_{U_1})^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L \cap (X_{M_{U_1}})^L$ whilst the maximal subtrees of $(U_2)^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L$ are $(A_{U_2})^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L \cap (X_{A_{U_2}})^L$ and $(M_{U_2})^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L \cap (X_{M_{U_2}})^L$. We will show that $(A_{U_1})^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L \cap (X_{A_{U_1}})^L = (A_{U_2})^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L \cap (X_{A_{U_2}})^L$ and $(M_{U_1})^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L \cap (X_{M_{U_1}})^L = (M_{U_2})^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L \cap (X_{M_{U_2}})^L$. Recall that $R_{\mathcal{T}_1} \cap X_{U_1}$ was defined as $(R_{\mathcal{T}_1} \cap X_{A_{U_1}}) \cup (R_{\mathcal{T}_1} \cap X_{M_{U_1}})$. Therefore $(R_{\mathcal{T}_1} \cap X_{U_1})^L = ((R_{\mathcal{T}_1} \cap X_{A_{U_1}}) \cup (R_{\mathcal{T}_1} \cap X_{M_{U_1}}))^L$. It is easy to show that $((R_{\mathcal{T}_1} \cap X_{A_{U_1}}) \cup (R_{\mathcal{T}_1} \cap X_{M_{U_1}}))^L = (R_{\mathcal{T}_1} \cap X_{A_{U_1}})^L \cup (R_{\mathcal{T}_1} \cap X_{M_{U_1}})^L$. Recall that $(R_{\mathcal{T}_1} \cap X_{A_{U_1}})^L = (X_{A_{U_1}})^L \cap (X_{A_{U_2}})^L$ and $(R_{\mathcal{T}_1} \cap X_{M_{U_1}})^L = (X_{M_{U_1}})^L \cap (X_{M_{U_2}})^L$. Hence $(R_{\mathcal{T}_1} \cap X_{U_1})^L = ((X_{A_{U_1}})^L \cap (X_{A_{U_2}})^L) \cup ((X_{M_{U_1}})^L \cap (X_{M_{U_2}})^L)$. Thus $(R_{\mathcal{T}_1} \cap X_{U_1})^L \cap (X_{A_{U_1}})^L = (X_{A_{U_1}})^L \cap (X_{A_{U_2}})^L$, as $(X_{M_{U_1}})^L \cap (X_{A_{U_1}})^L = \emptyset$. Likewise $(R_{\mathcal{T}_1} \cap X_{U_1})^L \cap (X_{A_{U_2}})^L = (X_{A_{U_1}})^L \cap (X_{A_{U_2}})^L$, as $(X_{M_{U_2}})^L \cap (X_{A_{U_2}})^L = \emptyset$. By the hypothesis of the proposition, $(A_{U_1})^L | ((X_{A_{U_1}})^L \cap (X_{A_{U_2}})^L) = (A_{U_2})^L | ((X_{A_{U_1}})^L \cap (X_{A_{U_2}})^L)$; hence $(A_{U_1})^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L \cap (X_{A_{U_1}})^L = (A_{U_2})^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L \cap (X_{A_{U_2}})^L$. Similarly, we can show that $(M_{U_1})^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L \cap (X_{M_{U_1}})^L = (M_{U_2})^L | (R_{\mathcal{T}_1} \cap X_{U_1})^L \cap (X_{M_{U_2}})^L$. Therefore $(U_1 | R_{\mathcal{T}_1} \cap X_{U_1})^L = (U_2 | W_{\mathcal{T}_2} \cap X_{U_2})^L$. Next we will show that $(U_2 | R_{\mathcal{T}_2} \cap X_{U_2})^L = (U_1 | W_{\mathcal{T}_1} \cap X_{U_1})^L$. Since $(R_{\mathcal{T}_2} \cap X_{U_2})^L = (W_{\mathcal{T}_1} \cap X_{U_1})^L$, showing that $(U_2)^L | (R_{\mathcal{T}_2} \cap X_{U_2})^L = (U_1)^L | (R_{\mathcal{T}_2} \cap X_{U_2})^L$ will suffice. We will show that $(A_{U_2})^L | (R_{\mathcal{T}_2} \cap X_{U_2})^L \cap (X_{A_{U_2}})^L = (M_{U_1})^L | (R_{\mathcal{T}_2} \cap X_{U_2})^L \cap (X_{M_{U_1}})^L$ and $(M_{U_2})^L | (R_{\mathcal{T}_2} \cap X_{U_2})^L \cap (X_{M_{U_2}})^L = (A_{U_1})^L | (R_{\mathcal{T}_2} \cap X_{U_2})^L \cap (X_{A_{U_1}})^L$. Now $(R_{\mathcal{T}_2} \cap X_{U_2})^L = ((X_{A_{U_2}})^L \cap (X_{M_{U_1}})^L) \cup ((X_{M_{U_2}})^L \cap (X_{A_{U_1}})^L)$. Hence $(R_{\mathcal{T}_2} \cap X_{U_2})^L \cap (X_{A_{U_2}})^L = (X_{A_{U_2}})^L \cap (X_{M_{U_1}})^L$, as $(X_{M_{U_2}})^L \cap (X_{A_{U_2}})^L = \emptyset$. We also have $(R_{\mathcal{T}_2} \cap X_{U_2})^L \cap (X_{M_{U_1}})^L = (X_{A_{U_2}})^L \cap (X_{M_{U_1}})^L$, as $(X_{M_{U_1}})^L \cap (X_{A_{U_1}})^L = \emptyset$. By the hypothesis of the proposition, we get the result. Similarly, $(M_{U_2})^L | (R_{\mathcal{T}_2} \cap X_{U_2})^L \cap (X_{M_{U_2}})^L = (A_{U_1})^L | (R_{\mathcal{T}_2} \cap X_{U_2})^L \cap (X_{A_{U_1}})^L$. Therefore $(U_2 | R_{\mathcal{T}_2} \cap X_{U_2})^L = (U_1 | W_{\mathcal{T}_1} \cap X_{U_1})^L$. Suppose that ITERATIVEALG returns the statement “displays a tree twice” upon input of (U_1, U_2) . We see that $(U_1 | R_{\mathcal{T}_1} \cap X_{U_1})^L = (U_2 | W_{\mathcal{T}_2} \cap X_{U_2})^L$ and $(U_2 | R_{\mathcal{T}_2} \cap X_{U_2})^L = (U_1 | W_{\mathcal{T}_1} \cap X_{U_1})^L$, by construction.

The final case is when (U_1, U_2) is a fertile major tree pair or a productive sterile major tree pair, and here we are going to need induction. We will use induction on the length of the longest maximal chain of fertile/(productive sterile) major tree pairs starting at any fertile major tree pair or productive sterile major tree pair. Base case: Let (U_1, U_2) be any fertile major tree pair such that the length of the longest maximal chain of fertile/(productive sterile) major tree

pairs starting at (U_1, U_2) is two. Let A_{U_1}, M_{U_1} be the two maximal subtrees of U_1 , and let A_{U_2}, M_{U_2} be the two maximal subtrees of U_2 . Without loss of generality, suppose that (A_{U_1}, A_{U_2}) and (M_{U_1}, M_{U_2}) are tree pairs. Both $(U_1, U_2), (A_{U_1}, A_{U_2})$ and $(U_1, U_2), (M_{U_1}, M_{U_2})$ are chains of tree pairs. Since the length of the longest maximal chain of fertile/(productive sterile) major tree pairs starting at (U_1, U_2) is two, (A_{U_1}, A_{U_2}) and (M_{U_1}, M_{U_2}) are not fertile major tree pairs nor productive sterile major tree pairs. Therefore we can have both being non-productive sterile major tree pairs or one being a trivial tree pair and the other being a non-productive sterile major tree pair. In both cases we can show that $(A_{U_1}|R_{\mathcal{T}_1} \cap X_{A_{U_1}})^L = (A_{U_2}|W_{\mathcal{T}_2} \cap X_{A_{U_2}})^L$, $(A_{U_2}|R_{\mathcal{T}_2} \cap X_{A_{U_2}})^L = (A_{U_1}|W_{\mathcal{T}_1} \cap X_{A_{U_1}})^L$, $(M_{U_1}|R_{\mathcal{T}_1} \cap X_{M_{U_1}})^L = (M_{U_2}|W_{\mathcal{T}_2} \cap X_{M_{U_2}})^L$, and $(M_{U_2}|R_{\mathcal{T}_2} \cap X_{M_{U_2}})^L = (M_{U_1}|W_{\mathcal{T}_1} \cap X_{M_{U_1}})^L$. Therefore $(U_1|R_{\mathcal{T}_1} \cap X_{U_1})^L = (U_2|W_{\mathcal{T}_2} \cap X_{U_2})^L$ and $(U_2|R_{\mathcal{T}_2} \cap X_{U_2})^L = (U_1|W_{\mathcal{T}_1} \cap X_{U_1})^L$. Similarly, if (U_1, U_2) is a productive sterile major tree pair then (U_1, U_2) yields a major tree pair (Y_1, Y_2) via ITERATIVEALG. Hence $(U_1, U_2), (Y_1, Y_2)$ is a chain of tree pairs. Since the length of the longest maximal chain of fertile/(productive sterile) major tree pairs starting at (U_1, U_2) is two, (Y_1, Y_2) must be a non-productive sterile major tree pair. We can show that $(Y_1|R_{\mathcal{T}_1} \cap X_{Y_1})^L = (Y_2|W_{\mathcal{T}_2} \cap X_{Y_2})^L$ and $(Y_2|R_{\mathcal{T}_2} \cap X_{Y_2})^L = (Y_1|W_{\mathcal{T}_1} \cap X_{Y_1})^L$. Moreover, it follows that $(Y_1)^L|(R_{\mathcal{T}_1})^L \cap (X_{Y_1})^L = (Y_2)^L|(R_{\mathcal{T}_1})^L \cap (X_{Y_2})^L$ and $(Y_2)^L|(R_{\mathcal{T}_2})^L \cap (X_{Y_2})^L = (Y_1)^L|(R_{\mathcal{T}_2})^L \cap (X_{Y_1})^L$; thus, by Proposition 5.7, $(U_1)^L|(R_{\mathcal{T}_1})^L \cap (X_{U_1})^L = (U_2)^L|(R_{\mathcal{T}_1})^L \cap (X_{U_2})^L$ and $(U_2)^L|(R_{\mathcal{T}_2})^L \cap (X_{U_2})^L = (U_1)^L|(R_{\mathcal{T}_2})^L \cap (X_{U_1})^L$. It follows that $(U_1|R_{\mathcal{T}_1} \cap X_{U_1})^L = (U_2|W_{\mathcal{T}_2} \cap X_{U_2})^L$ and $(U_2|R_{\mathcal{T}_2} \cap X_{U_2})^L = (U_1|W_{\mathcal{T}_1} \cap X_{U_1})^L$. Induction step: Suppose the result holds for any fertile/(productive sterile) major tree pair (Q_1, Q_2) such that the length of the longest maximal chain of fertile/(productive sterile) major tree pairs starting at (Q_1, Q_2) is less than or equal to p , where $p \geq 2$. Let (U_1, U_2) be any fertile major tree pair such that the length of the longest maximal chain of fertile major tree pairs starting at (U_1, U_2) is $p+1$. Let the two maximal subtrees of U_1 be A_{U_1}, M_{U_1} , and let the two maximal subtrees of U_2 be A_{U_2}, M_{U_2} . Without loss of generality, suppose that (A_{U_1}, A_{U_2}) and (M_{U_1}, M_{U_2}) are tree pairs. Since $p \geq 2$, (A_{U_1}, A_{U_2}) and (M_{U_1}, M_{U_2}) are fertile major tree pairs. Now the length of any maximal chain of fertile major tree pairs starting at (A_{U_1}, A_{U_2}) is less than or equal to p . Therefore the result holds for (A_{U_1}, A_{U_2}) , which means that $(A_{U_1}|R_{\mathcal{T}_1} \cap X_{A_{U_1}})^L = (A_{U_2}|W_{\mathcal{T}_2} \cap X_{A_{U_2}})^L$ and $(A_{U_2}|R_{\mathcal{T}_2} \cap X_{A_{U_2}})^L = (A_{U_1}|W_{\mathcal{T}_1} \cap X_{A_{U_1}})^L$. Likewise for (M_{U_1}, M_{U_2}) . Therefore $(U_1|R_{\mathcal{T}_1} \cap X_{U_1})^L = (U_2|W_{\mathcal{T}_2} \cap X_{U_2})^L$ and $(U_2|R_{\mathcal{T}_2} \cap X_{U_2})^L = (U_1|W_{\mathcal{T}_1} \cap X_{U_1})^L$. Similarly, if (Q_1, Q_2) is any productive sterile major tree pair then we get the result with the help of Proposition 5.7.

Coming back to \mathcal{T} and its maximal subtrees $\mathcal{T}_1, \mathcal{T}_2$. Now $(\mathcal{T}_1, \mathcal{T}_2)$ is a major tree pair of \mathcal{T} , and $(\mathcal{T}_1, \mathcal{T}_2)$ can be either a non-productive sterile major tree pair, a fertile major tree pair, or a

productive sterile major tree pair. In all cases we have $(\mathcal{T}_1|R_{\mathcal{T}_1} \cap X_{\mathcal{T}_1})^L = (\mathcal{T}_2|W_{\mathcal{T}_2} \cap X_{\mathcal{T}_2})^L$ and $(\mathcal{T}_2|R_{\mathcal{T}_2} \cap X_{\mathcal{T}_2})^L = (\mathcal{T}_1|W_{\mathcal{T}_1} \cap X_{\mathcal{T}_1})^L$. Therefore $(\mathcal{T}|R)^L = (\mathcal{T}|W)^L$. \square

Chapter 6

An Algorithm for Deciding whether or not a Basic Cyclic Pair Displays a Tree Twice

6.1 Introduction

In this chapter we present an efficient algorithm that decides whether or not a basic cyclic pair displays a tree twice. This algorithm uses, as its main subroutine, another efficient algorithm that decides whether or not a basic^0 representation displays a tree twice. Another subroutine that the algorithm uses is one that identifies basic cyclic pairs that never display a tree twice. If a basic cyclic pair is not represented by a basic^0 representation nor is a desirable configuration then it is of one of three forms. Each one of those three forms have almost all their basic^0 representations that are parsimoniously displayed being desirable configurations; hence the ones that are not desirable configurations, for which there are at most two, are sent to the main subroutine. In this way we get an efficient algorithm that decides whether or not a basic cyclic pair displays a tree twice.

6.2 An Efficient Algorithm for Deciding whether or not a Basic^0 Representation Displays a Tree Twice

Algorithm : BASIC^0 CHECKER

Input : A basic^0 representation \mathcal{T} representing the basic cyclic pair $B-(w, v, P_1, P_2)$ such that the maximal subtrees of \mathcal{T} are labeled $\mathcal{T}_1, \mathcal{T}_2$, and the maximal subtrees of \mathcal{T}_1 are labeled $A_{\mathcal{T}_1}, M_{\mathcal{T}_2}$ whilst the maximal subtrees of \mathcal{T}_2 are labeled $A_{\mathcal{T}_2}, M_{\mathcal{T}_2}$. We also have $v^{e_1} \in X_{M_{\mathcal{T}_1}}$ and $v^{e_2} \in X_{M_{\mathcal{T}_2}}$.

Output : The statement “displays a tree twice” or the statement “does not display a tree twice.”

1. Check if $(\mathcal{T}_1)^L = (\mathcal{T}_2)^L$. If so then go to Step 6.
2. Check if $(\mathcal{T}_2)^L \mid (X_{M_{\mathcal{T}_1}})^L = (M_{\mathcal{T}_1})^L$ and $C_{\mathcal{T}_2} = \emptyset$. If so then go to Step 6.
3. Check if $(\mathcal{T}_1)^L \mid (X_{M_{\mathcal{T}_2}})^L = (M_{\mathcal{T}_2})^L$ and $C_{\mathcal{T}_1} = \emptyset$. If so then go to Step 6.
4. Check if $C_{\mathcal{T}} = \emptyset$. If so then go to Step 5. If not then go to Step 7.
5. For every sterile major tree pair (Q_1, Q_2) such that (Q_1, Q_2) is the last tree pair of a maximal chain of fertile/(productive sterile) major tree pairs starting at $(\mathcal{T}_1, \mathcal{T}_2)$ check if

$$\begin{aligned} (M_{Q_1})^L \mid \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) &= (M_{Q_2})^L \mid \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right), \\ (A_{Q_1})^L \mid \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) &= (A_{Q_2})^L \mid \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right), \\ (M_{Q_2})^L \mid \left((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \right) &= (A_{Q_1})^L \mid \left((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \right), \quad \text{and} \\ (A_{Q_2})^L \mid \left((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \right) &= (M_{Q_1})^L \mid \left((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \right), \end{aligned}$$
 where A_{Q_1}, M_{Q_1} are the maximal subtrees of Q_1 and A_{Q_2}, M_{Q_2} are the maximal subtrees of Q_2 , or (Q_1, Q_2) meets the input criteria of `ITERATIVEALG` and `ITERATIVEALG` returns the statement “displays a tree twice” upon input of (Q_1, Q_2) . If so then go to Step 6. If not then go to Step 7.
6. Return the statement “displays a tree twice.”
7. Return the statement “does not display a tree twice.”

6.2.1 Proof that BASIC^0 CHECKER works

Theorem 6.1. *Let \mathcal{T} be a basic^0 representation. \mathcal{T} displays a tree twice if and only if BASIC^0 CHECKER returns the statement “displays a tree twice.”*

Proof. If BASIC^0 CHECKER returns the statement “displays a tree twice” then either $(\mathcal{T}_1)^L = (\mathcal{T}_2)^L$, $C_{\mathcal{T}_2} = \emptyset$ and $(\mathcal{T}_2)^L \mid (X_{M_{\mathcal{T}_1}})^L = (M_{\mathcal{T}_1})^L$, $C_{\mathcal{T}_1} = \emptyset$ and $(\mathcal{T}_1)^L \mid (X_{M_{\mathcal{T}_2}})^L = (M_{\mathcal{T}_2})^L$, or $C_{\mathcal{T}} = \emptyset$ and for every sterile major tree pair (Q_1, Q_2) such that (Q_1, Q_2) is the last tree pair of a maximal chain of fertile/(productive sterile) major tree pairs starting at $(\mathcal{T}_1, \mathcal{T}_2)$ either $(M_{Q_1})^L \mid \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) = (M_{Q_2})^L \mid \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right)$,

$(A_{Q_1})^L \mid \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) = (A_{Q_2})^L \mid \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right),$
 $(M_{Q_2})^L \mid \left((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \right) = (A_{Q_1})^L \mid \left((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \right),$ and
 $(A_{Q_2})^L \mid \left((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \right) = (M_{Q_1})^L \mid \left((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \right),$ where A_{Q_1}, M_{Q_1} are the maximal subtrees of Q_1 and A_{Q_2}, M_{Q_2} are the maximal subtrees of Q_2 , or (Q_1, Q_2) meets the input criteria of ITERATIVEALG and ITERATIVEALG returns the statement “displays a tree twice” upon input of (Q_1, Q_2) . In each case \mathcal{T} displays a tree twice, by Proposition 5.1, Proposition 5.2, Proposition 5.2, and Proposition 5.10, respectively.

If BASIC⁰ CHECKER returns the statement “does not display a tree twice” then there are two ways in which this can occur. The first way is when $(\mathcal{T}_1)^L \neq (\mathcal{T}_2)^L$, $C_{\mathcal{T}_2} \neq \emptyset$ or $(\mathcal{T}_2)^L \mid (X_{M_{\mathcal{T}_1}})^L \neq (M_{\mathcal{T}_1})^L$, $C_{\mathcal{T}_1} \neq \emptyset$ or $(\mathcal{T}_1)^L \mid (X_{M_{\mathcal{T}_2}})^L \neq (M_{\mathcal{T}_2})^L$, and $C_{\mathcal{T}} \neq \emptyset$. Now \mathcal{T} does not display a tree twice, because the only way that \mathcal{T} can display a tree twice when $(\mathcal{T}_1)^L \neq (\mathcal{T}_2)^L$, $C_{\mathcal{T}_2} \neq \emptyset$ or $(\mathcal{T}_2)^L \mid (X_{M_{\mathcal{T}_1}})^L \neq (M_{\mathcal{T}_1})^L$, and $(C_{\mathcal{T}_1} \neq \emptyset$ or $(\mathcal{T}_1)^L \mid (X_{M_{\mathcal{T}_2}})^L \neq (M_{\mathcal{T}_2})^L)$ is when $C_{\mathcal{T}} = \emptyset$. The second way in which the statement “does not display a tree twice” can be returned is when $(\mathcal{T}_1)^L \neq (\mathcal{T}_2)^L$, $(\mathcal{T}_2)^L \mid (X_{M_{\mathcal{T}_1}})^L \neq (M_{\mathcal{T}_1})^L$, $(\mathcal{T}_1)^L \mid (X_{M_{\mathcal{T}_2}})^L \neq (M_{\mathcal{T}_2})^L$, $C_{\mathcal{T}} = \emptyset$, and there exists a sterile major tree pair (Q_1, Q_2) that is the last tree pair of a maximal chain of fertile/(productive sterile) major tree pairs starting at $(\mathcal{T}_1, \mathcal{T}_2)$ such that at least one of the following does not hold: $(M_{Q_1})^L \mid \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right) = (M_{Q_2})^L \mid \left((X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L \right),$
 $(A_{Q_1})^L \mid \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right) = (A_{Q_2})^L \mid \left((X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L \right),$
 $(M_{Q_2})^L \mid \left((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \right) = (A_{Q_1})^L \mid \left((X_{M_{Q_2}})^L \cap (X_{A_{Q_1}})^L \right),$ and
 $(A_{Q_2})^L \mid \left((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \right) = (M_{Q_1})^L \mid \left((X_{A_{Q_2}})^L \cap (X_{M_{Q_1}})^L \right),$ where A_{Q_1}, M_{Q_1} are the maximal subtrees of Q_1 and A_{Q_2}, M_{Q_2} are the maximal subtrees of Q_2 , and either (Q_1, Q_2) does not meet the input criteria of ITERATIVEALG or ITERATIVEALG returns the statement “does not display a tree twice” upon input of (Q_1, Q_2) . By Lemma 5.10, either all of or all but one of the following is not equal to the empty set: $(X_{A_{Q_1}})^L \cap (X_{A_{Q_2}})^L$, $(X_{A_{Q_1}})^L \cap (X_{M_{Q_2}})^L$, $(X_{M_{Q_1}})^L \cap (X_{A_{Q_2}})^L$, $(X_{M_{Q_1}})^L \cap (X_{M_{Q_2}})^L$. In the former case, \mathcal{T} does not display a tree twice, by Theorem 5.3. In the latter case we employ ITERATIVEALG, and if (Q_1, Q_2) does not meet the input criteria of ITERATIVEALG then \mathcal{T} does not display a tree twice, by Theorem 5.4. If (Q_1, Q_2) meets the input criteria of ITERATIVEALG then we must have ITERATIVEALG returning the statement “does not display a tree twice” upon input of (Q_1, Q_2) . In which case suppose for contradiction that \mathcal{T} displays a tree twice. Then there exists two switchings R and W of \mathcal{T} such that $(\mathcal{T}|R)^L = (\mathcal{T}|W)^L$. Let $R_{\mathcal{T}_1}$ and $R_{\mathcal{T}_2}$ be two sets such that $R_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $R_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, $(R_{\mathcal{T}_1})^L \cup (R_{\mathcal{T}_2})^L = (R)^L$, and $(R_{\mathcal{T}_1})^L \cap (R_{\mathcal{T}_2})^L = \emptyset$. Likewise, let $W_{\mathcal{T}_1}$ and $W_{\mathcal{T}_2}$ be two sets

such that $W_{\mathcal{T}_1} \subseteq X_{\mathcal{T}_1}$, $W_{\mathcal{T}_2} \subseteq X_{\mathcal{T}_2}$, $(W_{\mathcal{T}_1})^L \cup (W_{\mathcal{T}_2})^L = (R)^L$, and $(W_{\mathcal{T}_1})^L \cap (W_{\mathcal{T}_2})^L = \emptyset$. Since $(\mathcal{T}_1)^L \neq (\mathcal{T}_2)^L$ and $C_{\mathcal{T}} = \emptyset$, we have $(\mathcal{T}_1|R_{\mathcal{T}_1})^L = (\mathcal{T}_2|W_{\mathcal{T}_2})^L$ and $(\mathcal{T}_2|R_{\mathcal{T}_2})^L = (\mathcal{T}_1|W_{\mathcal{T}_1})^L$. By Lemma 5.4, $(\mathcal{T}_1)^L|(R_{\mathcal{T}_1})^L = (\mathcal{T}_2)^L|(R_{\mathcal{T}_1})^L$ and $(\mathcal{T}_2)^L|(R_{\mathcal{T}_2})^L = (\mathcal{T}_1)^L|(R_{\mathcal{T}_2})^L$. By Lemma 5.5, $(Q_1)^L|(R_{\mathcal{T}_1})^L \cap (X_{Q_1})^L = (Q_2)^L|(R_{\mathcal{T}_1})^L \cap (X_{Q_2})^L$ and $(Q_2)^L|(R_{\mathcal{T}_2})^L \cap (X_{Q_2})^L = (Q_1)^L|(R_{\mathcal{T}_2})^L \cap (X_{Q_1})^L$. By Theorem 5.5, $(Q_1)^L|(R_{\mathcal{T}_p})^L \cap (X_{Q_1})^L \neq (Q_2)^L|(R_{\mathcal{T}_p})^L \cap (X_{Q_2})^L$, for some $p \in \{1, 2\}$ - a contradiction. Therefore \mathcal{T} does not display a tree twice. \square

6.2.2 Running-time of BASIC⁰ CHECKER

The checks that are made in BASIC⁰ CHECKER are checking whether or not two trees are isomorphic (in the phylogenetic sense), and checking whether or not two trees are isomorphic is linear-time [29]. The maximum number of steps that BASIC⁰ CHECKER can take is a polynomial in terms of the number of leaves in \mathcal{T} . This is the case because the number of maximal chains of fertile/(productive sterile) major tree pairs starting at $(\mathcal{T}_1, \mathcal{T}_2)$ is less than or equal to the number of leaves in \mathcal{T} . In order to get a maximal chain of fertile/(productive sterile) major tree pairs starting at $(\mathcal{T}_1, \mathcal{T}_2)$ and to carry out Step 5 in BASIC⁰ CHECKER, ITERATIVEALG may be needed. The maximum number of times that ITERATIVEALG can be employed is less than or equal to the number of leaves in \mathcal{T} . Moreover, we can find a one to one function from steps in BASIC⁰ CHECKER to the leaves in \mathcal{T} . Therefore BASIC⁰ CHECKER runs in time of $O(|X_{\mathcal{T}}|)$.

6.3 Checking a Basic Cyclic Pair

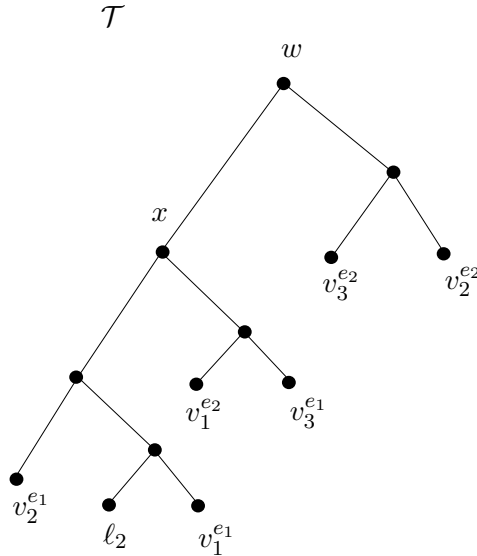
A basic cyclic pair is modelled by its tree representation. The tree representation parsimoniously displays basic⁰ representations. It would not be efficient to have to check every parsimoniously displayed basic⁰ representation; hence we must prove that almost all the basic⁰ representations that are parsimoniously displayed do not display a tree twice. We prove this with the help of two lemmas that describe desirable configurations. In fact, the desirable configurations extend to basic cyclic pairs, so a given basic cyclic pair may itself never display a tree twice, if it meets certain conditions. Therefore, based on these lemmas that describe desirable configurations, we can get an algorithm that will tell us when the basic cyclic pair is a desirable configuration or if it needs further checking. If the basic cyclic pair needs further checking then it must have not met the conditions of the desirable configuration lemmas. Hence the basic cyclic pair will meet some other set of conditions. These other conditions lead to an algorithm for checking whether or not a basic cyclic pair displays a tree twice.

6.3.1 Desirable configuration algorithm

Here we present an algorithm for identifying whether or not a basic cyclic pair meets certain conditions. If the basic cyclic pair does meet one of those conditions then it does not display a tree twice. If it does not meet any one of those conditions then it will need further checking. As another consequence of not meeting any of the conditions, the basic cyclic pair must be of one of three forms. We will describe those three forms in the next subsection; for this current subsection, we need a few more definitions. In a tree representation of a basic cyclic pair, we call two reticulation leaves partners when each has come from the same reticulation vertex. Given a basic cyclic pair with source v , we call a vertex x in the tree representing the basic cyclic pair a main ancestor when x is an ancestor of v^{e_1} or v^{e_2} in the tree representing the basic cyclic pair. When we say that a vertex x is an ancestor of partners what we mean is that there exists partners that are both descendants of x . Likewise when we say a vertex is an ancestor of a fixed leaf. Of course, a vertex is a main ancestor of partners when it is a main ancestor and an ancestor of partners. Likewise for a main ancestor of a fixed leaf.

Definition 6.1. Let \mathcal{T} be the representation of a basic cyclic pair $B-(w, v, P_1, P_2)$. Let v_i , where $i \in \mathbb{N}$, be a reticulation vertex in $B-(w, v, P_1, P_2)$. In \mathcal{T} , the reticulation leaves $v_i^{e_1}, v_i^{e_2}$ are called *partners*. The *partner* of $v_i^{e_1}$ is $v_i^{e_2}$, and the partner of $v_i^{e_2}$ is $v_i^{e_1}$.

Example 6.1. The reticulation leaves $v_1^{e_1}$ and $v_1^{e_2}$ are partners in the following figure.



Definition 6.2. Let \mathcal{T} be the representation of a basic cyclic pair $B-(w, v, P_1, P_2)$. We say that a vertex x in \mathcal{T} is a *main ancestor* when x is an ancestor of v^{e_1} or v^{e_2} .

Example 6.2. The vertex x from Example 6.1 is a main ancestor when \mathcal{T} represents the basic cyclic pair $B-(w, v_2, P_1, P_2)$. We also see that x is a main ancestor of partners as well as a main ancestor of a fixed leaf.

Algorithm : DESIRABLE CONFIG

Input : A tree \mathcal{T} , with root ρ , representing a basic cyclic pair $B-(w, v, P_1, P_2)$.

Output : The statement “does not display a tree twice” or the statement “needs further checking.”

Check if one of the following holds:

- i. Both children of ρ are ancestors of partners.
- ii. Both children of ρ are ancestors of a fixed leaf.
- iii. One of the children of ρ is an ancestor of a fixed leaf whilst the other is an ancestor of partners.
- iv. One of the children of ρ has a child that is a main ancestor of a fixed leaf.
- v. One of the children of ρ has one child being a main ancestor of partners whilst the other child is an ancestor of a reticulation leaf whose partner is a descendant of the other child of ρ .
- vi. One of the children of ρ has both its children being ancestors of partners.

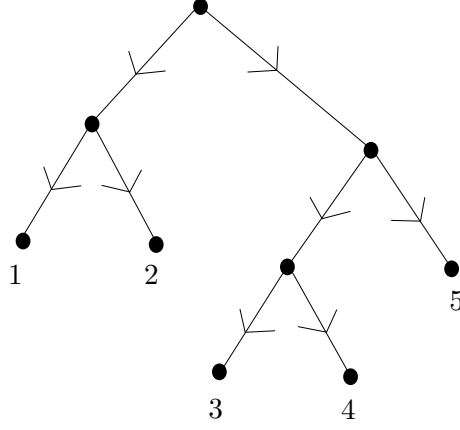
If any one of the above holds then return the statement “does not display a tree twice” otherwise return the statement “needs further checking.”

6.3.1.1 Proof that DESIRABLE CONFIG works

We need to prove that if one of the conditions from i) to vi) hold then the basic cyclic pair does not display a tree twice. In order to do that we need a few lemmas and a definition.

Definition 6.3. Let T be a tree on leaf set X . Let $x, y \in X$. We say that x and y *form a cherry* when x and y have the same parent.

Example 6.3. In the following figure we see that the leaves 1 and 2 form a cherry, as do the leaves 3 and 4.



Lemma 6.1. *Let \mathcal{T} be the tree representing a basic cyclic pair $B-(w, v, P_1, P_2)$. Let $\mathcal{T}_1, \mathcal{T}_2$ be the two maximal subtrees of \mathcal{T} . Let $v^{e_1} \in X_{\mathcal{T}_1}$ and $v^{e_2} \in X_{\mathcal{T}_2}$. Let ρ be the root of \mathcal{T} . If both children of ρ are ancestors of a fixed leaf then \mathcal{T} does not display a tree twice.*

Proof. Suppose both children of ρ are ancestors of a fixed leaf then there exists a fixed leaf ℓ_1 such that $\ell_1 \in X_{\mathcal{T}_1}$ and there exists a fixed leaf ℓ_2 such that $\ell_2 \in X_{\mathcal{T}_2}$. Let Γ be the reduction of $B-(w, v, P_1, P_2)$ where $\Gamma = \{v^{e_1}, v^{e_2}, \ell_1, \ell_2\}$. Now $\mathcal{T}|\Gamma$ has only two switchings, namely $R_1 = \{v^{e_1}, \ell_1, \ell_2\}$ and $R_2 = \{v^{e_2}, \ell_1, \ell_2\}$. We see that $((\mathcal{T}|\Gamma)|R_1)^L$ has the triple $\ell_v \ell_1 | \ell_2$ whilst $((\mathcal{T}|\Gamma)|R_2)^L$ has the triple $\ell_v \ell_2 | \ell_1$, so, by Lemma 1.2, $((\mathcal{T}|\Gamma)|R_1)^L \neq ((\mathcal{T}|\Gamma)|R_2)^L$. Therefore $\mathcal{T}|\Gamma$ does not display a tree twice. Thus, by Corollary 4.1, $B-(w, v, P_1, P_2)$ does not display a tree twice. Therefore, by Theorem 4.1, \mathcal{T} does not display a tree twice. \square

Lemma 6.2. *Let \mathcal{T} be the tree representing a basic cyclic pair $B-(w, v, P_1, P_2)$. Let $\mathcal{T}_1, \mathcal{T}_2$ be the two maximal subtrees of \mathcal{T} . Let $v^{e_1} \in X_{\mathcal{T}_1}$ and $v^{e_2} \in X_{\mathcal{T}_2}$. Let ρ be the root of \mathcal{T} . If one child of ρ has both its children being ancestors of a fixed leaf then \mathcal{T} does not display a tree twice.*

Proof. Without loss of generality, suppose that the child of ρ that has both its children being ancestors of a fixed leaf is the root of \mathcal{T}_1 . Let $A_{\mathcal{T}_1}$ and $M_{\mathcal{T}_1}$ be the two maximal subtrees of \mathcal{T}_1 . Let $v^{e_1} \in X_{M_{\mathcal{T}_1}}$. Then there exists fixed leaves ℓ_1, ℓ_2 such that $\ell_1 \in X_{A_{\mathcal{T}_1}}$ and $\ell_2 \in X_{M_{\mathcal{T}_1}}$. Let Γ be the reduction of $B-(w, v, P_1, P_2)$ where $\Gamma = \{v^{e_1}, v^{e_2}, \ell_1, \ell_2\}$. Now $\mathcal{T}|\Gamma$ has only two switchings, namely $R_1 = \{v^{e_1}, \ell_1, \ell_2\}$ and $R_2 = \{v^{e_2}, \ell_1, \ell_2\}$. We see that $((\mathcal{T}|\Gamma)|R_1)^L$ has the triple $\ell_v \ell_2 | \ell_1$ whilst $((\mathcal{T}|\Gamma)|R_2)^L$ has the triple $\ell_1 \ell_2 | \ell_v$, so, by Lemma 1.2, $((\mathcal{T}|\Gamma)|R_1)^L \neq ((\mathcal{T}|\Gamma)|R_2)^L$. Therefore $\mathcal{T}|\Gamma$ does not display a tree twice. Thus, by Corollary 4.1, $B-(w, v, P_1, P_2)$ does not display a tree twice. Therefore, by Theorem 4.1, \mathcal{T} does not display a tree twice. \square

Proposition 6.1. *Let \mathcal{T} be the tree representing a basic cyclic pair $B-(w, v, P_1, P_2)$. Let $\mathcal{T}_1, \mathcal{T}_2$ be the two maximal subtrees of \mathcal{T} . Let $v^{e_1} \in X_{\mathcal{T}_1}$ and $v^{e_2} \in X_{\mathcal{T}_2}$. Let ρ be the root of \mathcal{T} . If one of the children of ρ has a child that is a main ancestor of a fixed leaf then \mathcal{T} does not display a tree twice.*

Proof. Without loss of generality, suppose that the root of \mathcal{T}_1 , call it ρ_1 , has a child that is a main ancestor of a fixed leaf. Let x_1, x_2 be ρ_1 's children. Let $A_{\mathcal{T}_1}$ and $M_{\mathcal{T}_1}$ be the two maximal subtrees of \mathcal{T}_1 , and let $v^{e_1} \in X_{M_{\mathcal{T}_1}}$. Let x_1 be the root of $A_{\mathcal{T}_1}$ and let x_2 be the root of $M_{\mathcal{T}_1}$. We must have x_2 being the child of ρ_1 that is a main ancestor of a fixed leaf; hence there exists a leaf $\ell \in X_{M_{\mathcal{T}_1}}$. If $X_{A_{\mathcal{T}_1}}$ contains a fixed leaf then, by Lemma 6.2, \mathcal{T} does not display a tree twice. Suppose $X_{A_{\mathcal{T}_1}}$ does not contain a fixed leaf then, since $X_{A_{\mathcal{T}_1}} \neq \emptyset$, there exists a reticulation leaf $v_1^{e_m} \in X_{A_{\mathcal{T}_1}}$, where $m \in \{1, 2\}$. Consider the reduction Γ of $B-(w, v, P_1, P_2)$ where $\Gamma = \{v_1^{e_1}, v_1^{e_2}, v^{e_1}, v^{e_2}, \ell\}$. We will prove that $\mathcal{T}|\Gamma$ does not display a tree twice. Consider the two switchings R_1, R_2 of $\mathcal{T}|\Gamma$ where $R_1 = \{v^{e_1}, v_1^{e_i}, \ell\}$ and $R_2 = \{v^{e_2}, v_1^{e_j}, \ell\}$ where $i, j \in \{1, 2\}$. We see that $((\mathcal{T}|\Gamma)|R_1)^L$ has either the triple $\ell\ell_v|\ell_{v_1}$ or $\ell_v\ell_{v_1}|\ell$, depending upon where $v_1^{e_1}$ and $v_1^{e_2}$ are in $\mathcal{T}|\Gamma$. If one of $v_1^{e_1}$ or $v_1^{e_2}$ form a cherry with v^{e_1} then $((\mathcal{T}|\Gamma)|R_1)^L$ has the triple $\ell_v\ell_{v_1}|\ell$, but in that case $((\mathcal{T}|\Gamma)|R_2)^L$ must have the triple $\ell_{v_1}\ell|\ell_v$. If neither $v_1^{e_1}$ nor $v_1^{e_2}$ form a cherry with v^{e_1} then we must have ℓ and v^{e_1} forming a cherry in $\mathcal{T}|\Gamma$. In that case $((\mathcal{T}|\Gamma)|R_1)^L$ has the triple $\ell_v\ell|\ell_{v_1}$ whilst $((\mathcal{T}|\Gamma)|R_2)^L$ has either $\ell\ell_{v_1}|\ell_v$ or $\ell_v\ell_{v_1}|\ell$. Therefore, by Lemma 1.2, $\mathcal{T}|\Gamma$ does not display a tree twice. Thus, by Corollary 4.1, $B-(w, v, P_1, P_2)$ does not display a tree twice. Therefore, by Theorem 4.1, \mathcal{T} does not display a tree twice. \square

Proposition 6.2. *Let \mathcal{T} be the tree representing a basic cyclic pair $B-(w, v, P_1, P_2)$. Let ρ be the root of \mathcal{T} . If one of the children of ρ has a child that is a main ancestor of partners whilst the other is an ancestor of a reticulation leaf, say $v_1^{e_m}$, where $m \in \{1, 2\}$, such that $v_1^{e_n}$, where $n \in \{1, 2\} \setminus \{m\}$, is a descendant of the other child of ρ then \mathcal{T} does not display a tree twice.*

Proof. We will show that every parsimoniously displayed basic⁰ representation does not display a tree twice, which, by Theorem 5.2, shows that \mathcal{T} does not display a tree twice. Let \mathcal{T}^* be any basic⁰ representation that is parsimoniously displayed by \mathcal{T} . Let \mathcal{T}_1^* and \mathcal{T}_2^* be the two maximal subtrees of \mathcal{T}^* . Without loss of generality, suppose that $v^{e_1} \in X_{\mathcal{T}_1^*}$. Let $A_{\mathcal{T}_1^*}$ and $M_{\mathcal{T}_1^*}$ be the two maximal subtrees of \mathcal{T}_1^* . Without loss of generality, suppose that $v^{e_1} \in X_{M_{\mathcal{T}_1^*}}$. Since \mathcal{T}^* is a parsimoniously displayed basic⁰ representation, we have $\{v_1^{e_1}, v_1^{e_2}\} \subseteq X_{\mathcal{T}^*}$, and it follows from the hypothesis that $\{v_1^{e_1}, v_1^{e_2}\} \cap X_{A_{\mathcal{T}_1^*}} \neq \emptyset$. Therefore there exists a fixed leaf ℓ such that $\ell \in X_{M_{\mathcal{T}_1^*}}$, thus the root of $M_{\mathcal{T}_1^*}$ is a main ancestor of a fixed leaf. Hence, by Proposition 6.1, \mathcal{T}^* does not display a tree twice. Therefore \mathcal{T} does not display a tree twice. \square

Proposition 6.3. *Let \mathcal{T} be the tree representing a basic cyclic pair $B-(w, v, P_1, P_2)$. Let ρ be the root of \mathcal{T} . If one of the children of ρ has both its children being ancestors of partners then \mathcal{T} does not display a tree twice.*

Proof. We will show that every parsimoniously displayed basic^0 representation does not display a tree twice, which, by Theorem 5.2, shows that \mathcal{T} does not display a tree twice. Let \mathcal{T}^* be any basic^0 representation that is parsimoniously displayed by \mathcal{T} . Let \mathcal{T}_1^* and \mathcal{T}_2^* be the two maximal subtrees of \mathcal{T}^* . We must have $C_{\mathcal{T}_1^*}$ or $C_{\mathcal{T}_2^*}$ being non-empty. Without loss of generality, suppose that $C_{\mathcal{T}_1^*} \neq \emptyset$. Let $A_{\mathcal{T}_1^*}$ and $M_{\mathcal{T}_1^*}$ be the two maximal subtrees of \mathcal{T}_1^* . Since \mathcal{T}^* is a parsimoniously displayed basic^0 representation, we must have $C_{A_{\mathcal{T}_1^*}} \neq \emptyset$ and $C_{M_{\mathcal{T}_1^*}} \neq \emptyset$. Therefore, by Lemma 6.2, \mathcal{T}^* does not display a tree twice. Therefore \mathcal{T} does not display a tree twice. \square

Theorem 6.2. *If DESIRABLE CONFIG returns the statement “does not display a tree twice” then the representation \mathcal{T} does not display a tree twice.*

Proof. If DESIRABLE CONFIG returns the statement “does not display a tree twice” then one of the conditions i) to vi) is present in \mathcal{T} . If condition i) holds then, by Lemma 5.1, \mathcal{T} does not display a tree twice. If condition ii) holds then, by Lemma 6.1, \mathcal{T} does not display a tree twice. If condition iii) holds then, by Lemma 5.2, \mathcal{T} does not display a tree twice. If condition iv) holds then, by Proposition 6.1, \mathcal{T} does not display a tree twice. If condition v) holds then, by Proposition 6.2, \mathcal{T} does not display a tree twice. If condition vi) holds then, by Proposition 6.3, \mathcal{T} does not display a tree twice. Therefore \mathcal{T} does not display a tree twice. \square

6.3.2 A representation that needs further checking

Given a representation of a basic cyclic pair such that having gone through the DESIRABLE CONFIG algorithm the output was “needs further checking,” we look at what this means in terms of the form of the representation. The only way that an output of “needs further checking” can occur is when the representation satisfies none of the conditions i) to vi) of the DESIRABLE CONFIG algorithm. This can occur when the representation is a basic^0 representation, in which case we can send the representation to BASIC⁰ CHECKER. Suppose the representation is not a basic^0 representation, and let ρ be the root of the representation, then one of the children of ρ is an ancestor of partners, but we can say more, namely that exactly one child of ρ is an ancestor of partners. Indeed, the other child of ρ is neither an ancestor of partners nor an ancestor of a fixed leaf. Now we can say something about the children of the child of ρ that is an ancestor of partners; exactly one could be an ancestor of a fixed leaf and/or an ancestor of partners but it can not be a main ancestor of a fixed leaf, and if it is a main ancestor of partners then its sibling can not be an ancestor of a reticulation leaf whose partners is a descendant of the other child of ρ ; hence all the descendants of the sibling of the main ancestor of partners are reticulation leaves whose partners are descendants of the main ancestor of partners.

Lemma 6.3. *Let \mathcal{T} be the tree representing a basic cyclic pair $B-(w, v, P_1, P_2)$. Let ρ be the root of \mathcal{T} . If DESIRABLE CONFIG outputs “needs further checking” and \mathcal{T} is not a basic^0 representation then exactly one of ρ ’s children, say x , is an ancestor of partners and x ’s sibling is not an ancestor of a fixed leaf. The children of x can have the following properties:*

1. *Both are neither ancestors of partners nor ancestors of fixed leaves.*
2. *The non-main ancestor is an ancestor of a fixed leaf and/or an ancestor of partners but is not a main ancestor of a fixed leaf, whilst the other child is not an ancestor of partners nor an ancestor of a fixed leaf*
3. *Exactly one is a main ancestor of partners and not an ancestor of a fixed leaf, and all the descendants of the other child are reticulation leaves whose partners are descendants of the main ancestor of partners.*

Proof. See the above paragraph. □

6.3.3 Checking a representation that needs further checking

If the representation that needs further checking is a basic^0 representation then we send it to BASIC^0 CHECKER. If the representation is not a basic^0 representation then it has one of three forms, based upon the children of the one child of the root that is an ancestor of partners. The first form that the representation can take is one in which both children are neither ancestors of partners nor ancestors of fixed leaves. In this case all basic^0 representations parsimoniously displayed by the representation except two never display a tree twice, and the two basic^0 representations that may display a tree twice are easily identifiable. The second form that the representation can take is one in which the non-main ancestor is an ancestor of a fixed leaf and/or an ancestor of partners but is not a main ancestor of a fixed leaf, whilst the other child is not an ancestor of partners nor an ancestor of a fixed leaf. In this case all basic^0 representations parsimoniously displayed by the representation except one never display a tree twice, so we only need to check one basic^0 representation in this case. The third form that a representation can take is one in which exactly one child is a main ancestor of partners and not an ancestor of a fixed leaf, whilst all the descendants of the other child are reticulation leaves whose partners are descendants of the main ancestor of partners. In this final case all the representations, except one, that are displayed via a general switching that only switches reticulation leaves that are descendants of the non-main ancestor of partners are desirable configurations. The one representation that possibly is not a desirable configuration has a vertex that

is an ancestor of a fixed leaf, so if that is not a desirable configuration then it is of the second form.

Algorithm : FORM 1) CHECKER

Input : A tree \mathcal{T} , with root ρ , representing a basic cyclic pair $B-(w, v, P_1, P_2)$. The maximal subtrees of \mathcal{T} are labeled $\mathcal{T}_1, \mathcal{T}_2$, and the maximal subtrees of \mathcal{T}_1 are labeled $A_{\mathcal{T}_1}, M_{\mathcal{T}_1}$, whilst the maximal subtrees of \mathcal{T}_2 are labeled $A_{\mathcal{T}_2}, M_{\mathcal{T}_2}$. We also have $v^{e_1} \in X_{M_{\mathcal{T}_1}}$ and $v^{e_2} \in X_{M_{\mathcal{T}_2}}$. In addition, exactly one child of ρ , without loss of generality let it be the root of \mathcal{T}_1 , which we call x , is an ancestor of partners whilst the other child of ρ is not an ancestor of a fixed leaf. No child of x is an ancestor of a fixed leaf nor an ancestor of partners.

Output : The statement “displays a tree twice” or the statement “does not display a tree twice.”

1. Send the tree $(\mathcal{T}|G_1)^L$ to BASIC⁰ CHECKER where $G_1 = X_{\mathcal{T}_2} \cup \left(X_{\mathcal{T}_1} \setminus \left\{ v_i^{e_m} \in X_{A_{\mathcal{T}_1}} \mid \ell_{v_i} \in (X_{A_{\mathcal{T}_1}})^L \cap (X_{M_{\mathcal{T}_1}})^L \right\} \right)$, where $m \in \{1, 2\}$ and $i \in \mathbb{N}$. If the statement “does not display a tree twice” is outputted from BASIC⁰ CHECKER then go to Step 2. Otherwise return the statement “displays a tree twice.”
2. Send the tree $(\mathcal{T}|G_2)^L$ to BASIC⁰ CHECKER where $G_2 = X_{\mathcal{T}_2} \cup \left(X_{\mathcal{T}_1} \setminus \left\{ v_i^{e_m} \in X_{M_{\mathcal{T}_1}} \mid \ell_{v_i} \in (X_{A_{\mathcal{T}_1}})^L \cap (X_{M_{\mathcal{T}_1}})^L \right\} \right)$. Output the statement outputted by BASIC⁰ CHECKER.

Proposition 6.4. *Let \mathcal{T} be the representation of the basic cyclic pair $B-(w, v, P_1, P_2)$. Let the root of \mathcal{T} be ρ . Let the maximal subtrees of \mathcal{T} be labeled $\mathcal{T}_1, \mathcal{T}_2$, and let the maximal subtrees of \mathcal{T}_1 be labeled $A_{\mathcal{T}_1}, M_{\mathcal{T}_1}$. Similarly, let the maximal subtrees of \mathcal{T}_2 be labeled $A_{\mathcal{T}_2}, M_{\mathcal{T}_2}$. We also have $v^{e_1} \in X_{M_{\mathcal{T}_1}}$ and $v^{e_2} \in X_{M_{\mathcal{T}_2}}$. In addition, exactly one child of ρ , without loss of generality let it be the root of \mathcal{T}_1 , which we call x , is an ancestor of partners whilst the other child of ρ is not an ancestor of a fixed leaf. No child of x is an ancestor of a fixed leaf nor an ancestor of partners. If FORM 1) CHECKER returns the statement “does not display a tree twice” then \mathcal{T} does not display a tree twice, and if FORM 1) CHECKER returns the statement “displays a tree twice” then \mathcal{T} displays a tree twice.*

Proof. If the statement “displays a tree twice” is returned then, since $(\mathcal{T}|G_1)^L$ and $(\mathcal{T}|G_2)^L$ are both basic⁰ representations parsimoniously displayed by \mathcal{T} , \mathcal{T} displays a tree twice, by Theorem 5.2. If the statement “does not display a tree twice” is outputted then both $(\mathcal{T}|G_1)^L$ and $(\mathcal{T}|G_2)^L$ do not display a tree twice. We now want to show that every other basic⁰ representation parsimoniously displayed by \mathcal{T} does not display a tree twice. Take any basic⁰ representation \mathcal{T}^* parsimoniously

displayed by \mathcal{T} . Then $\mathcal{T}^* = (\mathcal{T}|G)^L$ where G is a general switching such that $G = X_{\mathcal{T}_2} \cup (X_{\mathcal{T}_1} \setminus G^*)$ where $G^* \subseteq Q = \left\{ v_i^{e_m} \in X_{\mathcal{T}_1} \mid \ell_{v_i} \in (X_{A_{\mathcal{T}_1}})^L \cap (X_{M_{\mathcal{T}_1}})^L \right\}$, for $m \in \{1, 2\}$ and $i \in \mathbb{N}$, and for all $v_i^{e_m} \in Q$ exactly one of $v_i^{e_1}, v_i^{e_2}$ is in G^* . If $G = G_1$ or $G = G_2$ then we are done, so suppose $G \neq G_1$ and $G \neq G_2$ then there exists $v_j^{e_m} \in G^*$, where $j \in \mathbb{N}$, such that $v_j^{e_m} \in X_{M_{\mathcal{T}_1}}$ and $\ell_{v_j} \in (X_{A_{\mathcal{T}_1}})^L \cap (X_{M_{\mathcal{T}_1}})^L$, and there exists $v_i^{e_n} \in G^*$, where $n \in \{1, 2\}$, such that $v_i^{e_n} \in X_{A_{\mathcal{T}_1}}$ and $\ell_{v_i} \in (X_{A_{\mathcal{T}_1}})^L \cap (X_{M_{\mathcal{T}_1}})^L$. Therefore the root of the representation \mathcal{T}^* has a child whose children are both ancestors of a fixed leaf. Hence, by Lemma 6.2, \mathcal{T}^* never displays a tree twice. Therefore, by Theorem 5.2, \mathcal{T} never displays a tree twice. \square

Algorithm : FORM II) CHECKER

Input : A tree \mathcal{T} , with root ρ , representing a basic cyclic pair $B-(w, v, P_1, P_2)$. The maximal subtrees of \mathcal{T} are labeled $\mathcal{T}_1, \mathcal{T}_2$, and the maximal subtrees of \mathcal{T}_1 are labeled $A_{\mathcal{T}_1}, M_{\mathcal{T}_1}$, whilst the maximal subtrees of \mathcal{T}_2 are labeled $A_{\mathcal{T}_2}, M_{\mathcal{T}_2}$. We also have $v^{e_1} \in X_{M_{\mathcal{T}_1}}$ and $v^{e_2} \in X_{M_{\mathcal{T}_2}}$. Exactly one child of ρ , without loss of generality let it be the root of \mathcal{T}_1 , which we call x , is an ancestor of partners whilst the other child of ρ is not an ancestor of a fixed leaf. In addition, the root of $A_{\mathcal{T}_1}$ is an ancestor of partners or an ancestor of a fixed leaf, whilst the root of $M_{\mathcal{T}_1}$ is neither.

Output : The statement “displays a tree twice” or the statement “does not display a tree twice.”

1. Check if $(\mathcal{T}|G_1)^L$ is a basic^0 representation, where $G_1 = X_{\mathcal{T}_2} \cup \left(X_{\mathcal{T}_1} \setminus \left\{ v_i^{e_m} \in X_{M_{\mathcal{T}_1}} \mid \ell_{v_i} \in (X_{A_{\mathcal{T}_1}})^L \cap (X_{M_{\mathcal{T}_1}})^L \right\} \right)$, where $m \in \{1, 2\}$ and $i \in \mathbb{N}$. If it is then send it to BASIC^0 CHECKER, and return output.
2. If $(\mathcal{T}|G_1)^L$ is not a basic^0 representation then take any basic^0 representation parsimoniously displayed by $(\mathcal{T}|G_1)^L$ and send it to BASIC^0 CHECKER. Return output from BASIC^0 CHECKER.

Proposition 6.5. *Let \mathcal{T} be the representation of the basic cyclic pair $B-(w, v, P_1, P_2)$. Let the root of \mathcal{T} be ρ . Let the maximal subtrees of \mathcal{T} be labeled $\mathcal{T}_1, \mathcal{T}_2$, and let the maximal subtrees of \mathcal{T}_1 be labeled $A_{\mathcal{T}_1}, M_{\mathcal{T}_1}$. Similarly, let the maximal subtrees of \mathcal{T}_2 be labeled $A_{\mathcal{T}_2}, M_{\mathcal{T}_2}$. We also have $v^{e_1} \in X_{M_{\mathcal{T}_1}}$ and $v^{e_2} \in X_{M_{\mathcal{T}_2}}$. The root of $A_{\mathcal{T}_1}$ is an ancestor of partners or an ancestor of a fixed leaf, whilst the root of $M_{\mathcal{T}_1}$ is neither. In addition, the root of \mathcal{T}_1 is an ancestor of partners, whilst the root of \mathcal{T}_2 is not an ancestor of a fixed leaf nor an ancestor of partners. If FORM II) CHECKER returns the statement “does not display a tree twice” then \mathcal{T} does not display a tree twice, and if FORM II) CHECKER returns the statement “displays a tree twice” then \mathcal{T} displays a tree twice.*

Proof. If $(\mathcal{T}|G_1)^L$ is a basic^0 representation then we must have the root of $A_{\mathcal{T}_1}$ being an ancestor of a fixed leaf and not an ancestor of partners whilst the root of \mathcal{T}_1 is an ancestor of partners. Hence a similar (and shorter) proof to that of the proof that FORM I) CHECKER works can be used in that case. Suppose $(\mathcal{T}|G_1)^L$ is not a basic^0 representation then the root of $A_{\mathcal{T}_1}$ is an ancestor of partners. If we take any general switching of form $G = X_{\mathcal{T}_2} \cup (X_{\mathcal{T}_1} \setminus G^*)$ where $G^* \subseteq Q = \left\{ v_i^{e_m} \in X_{\mathcal{T}_1} \mid \ell_{v_i} \in (X_{A_{\mathcal{T}_1}})^L \cap (X_{M_{\mathcal{T}_1}})^L \right\}$, where $m \in \{1, 2\}$ and $i \in \mathbb{N}$, and for all $v_i^{e_m} \in Q$ exactly one of $v_i^{e_1}, v_i^{e_2}$ is in G^* , then, provided $G \neq G_1$, $(\mathcal{T}|G)^L$ can be reduced to a desirable configuration. Hence \mathcal{T} displays a tree twice if and only if $(\mathcal{T}|G_1)^L$ displays a tree twice. Since $(\mathcal{T}|G_1)^L$ is not a basic^0 representation, if there exists a basic^0 representation parsimoniously displayed by $(\mathcal{T}|G_1)^L$ that displays a tree twice then $(\mathcal{T}|G_1)^L$ displays a tree twice. Take any basic^0 representation \mathcal{T}^* parsimoniously displayed by $(\mathcal{T}|G_1)^L$ then it has the following form: exactly one of its maximal subtrees, say \mathcal{T}_1^* , has a fixed leaf. Moreover, exactly one of the children of the root of \mathcal{T}_1^* is an ancestor of a fixed leaf and is not a major ancestor of a fixed leaf. If BASIC^0 CHECKER returns “does not display a tree twice” when \mathcal{T}^* is inputted then \mathcal{T}^* does not display a tree twice, and every other basic^0 representation parsimoniously displayed by $(\mathcal{T}|G_1)^L$ will return “does not display a tree twice” when inputted into BASIC^0 CHECKER (see BASIC^0 CHECKER algorithm). Hence if “does not display a tree twice” is outputted then every basic^0 representation parsimoniously displayed by $(\mathcal{T}|G_1)^L$ does not display a tree twice. Thus $(\mathcal{T}|G_1)^L$ does not display a tree twice, which implies that \mathcal{T} does not display a tree twice. If the statement “displays a tree twice” is outputted then $(\mathcal{T}|G_1)^L$ displays a tree twice, and this implies that \mathcal{T} displays a tree twice. \square

Algorithm : FORM III) CHECKER

Input : A tree \mathcal{T} , with root ρ , representing a basic cyclic pair $B-(w, v, P_1, P_2)$. The maximal subtrees of \mathcal{T} are labeled $\mathcal{T}_1, \mathcal{T}_2$, and the maximal subtrees of \mathcal{T}_1 are labeled $A_{\mathcal{T}_1}, M_{\mathcal{T}_1}$, whilst the maximal subtrees of \mathcal{T}_2 are labeled $A_{\mathcal{T}_2}, M_{\mathcal{T}_2}$. We also have $v^{e_1} \in X_{M_{\mathcal{T}_1}}$ and $v^{e_2} \in X_{M_{\mathcal{T}_2}}$. The root of \mathcal{T}_1 is an ancestor of partners, and the root of $M_{\mathcal{T}_1}$ is a main ancestor of partners, whilst the root of $A_{\mathcal{T}_1}$ is neither an ancestor of partners nor an ancestor of a fixed leaf. There is also no descendant reticulation leaf of the root of $A_{\mathcal{T}_1}$ that has its partner as a descendant of the root of \mathcal{T}_2 . In addition, the root of \mathcal{T}_2 is neither an ancestor of partners nor an ancestor of a fixed leaf.

Output : The statement “displays a tree twice” or the statement “does not display a tree twice.”

1. Take $(\mathcal{T}|G_1)^L$, where $G_1 = X_{\mathcal{T}_2} \cup \left(X_{\mathcal{T}_1} \setminus \left\{ v_i^{e_m} \in X_{A_{\mathcal{T}_1}} \mid \ell_{v_i} \in (X_{A_{\mathcal{T}_1}})^L \cap (X_{M_{\mathcal{T}_1}})^L \right\} \right)$, for $m \in \{1, 2\}$ and $i \in \mathbb{N}$, and send it to DESIRABLE CONFIG. If we get the statement “needs further checking” then send $(\mathcal{T}|G_1)^L$ to FORM II) CHECKER, and return output. Otherwise return

output from DESIRABLE CONFIG.

Proposition 6.6. *Let \mathcal{T} be the representation of the basic cyclic pair $B-(w, v, P_1, P_2)$. Let the root of \mathcal{T} be ρ . Let the maximal subtrees of \mathcal{T} be labeled $\mathcal{T}_1, \mathcal{T}_2$, and let the maximal subtrees of \mathcal{T}_1 be labeled $A_{\mathcal{T}_1}, M_{\mathcal{T}_1}$. Similarly, let the maximal subtrees of \mathcal{T}_2 be labeled $A_{\mathcal{T}_2}, M_{\mathcal{T}_2}$. We also have $v^{e_1} \in X_{M_{\mathcal{T}_1}}$ and $v^{e_2} \in X_{M_{\mathcal{T}_2}}$. The root of \mathcal{T}_1 is an ancestor of partners, and the root of $M_{\mathcal{T}_1}$ is a main ancestor of partners, whilst the root of $A_{\mathcal{T}_1}$ is neither an ancestor of partners nor an ancestor of a fixed leaf. There is also no descendant reticulation leaf of the root of $A_{\mathcal{T}_1}$ that has its partner as a descendant of the root of \mathcal{T}_2 . In addition, the root of \mathcal{T}_2 is neither an ancestor of partners nor an ancestor of a fixed leaf. If FORM III) CHECKER returns the statement “does not display a tree twice” then \mathcal{T} does not display a tree twice, and if FORM III) CHECKER returns the statement “displays a tree twice” then \mathcal{T} displays a tree twice.*

Proof. Let G be any general switching of form $G = X_{\mathcal{T}_2} \cup (X_{\mathcal{T}_1} \setminus G^*)$ where $G^* \subseteq Q = \left\{ v_i^{e_m} \in X_{\mathcal{T}_1} \mid \ell_{v_i} \in (X_{A_{\mathcal{T}_1}})^L \cap (X_{M_{\mathcal{T}_1}})^L \right\}$, where $m \in \{1, 2\}$ and $i \in \mathbb{N}$, and for all $v_i^{e_m} \in Q$ exactly one of $v_i^{e_1}, v_i^{e_2}$ is in G^* . We see that if $G \neq G_1$ then $(\mathcal{T}|G)^L$ never displays a tree twice. Therefore \mathcal{T} displays a tree twice if and only if $(\mathcal{T}|G_1)^L$ displays a tree twice. If the statement “does not display a tree twice” is returned then $(\mathcal{T}|G_1)^L$ does not display a tree twice, by Proposition 6.5 or Theorem 6.2. If the statement “displays a tree twice” is returned then $(\mathcal{T}|G_1)^L$ displays a tree twice, by Proposition 6.5. Therefore if FORM III) CHECKER returns the statement “does not display a tree twice” then \mathcal{T} does not display a tree twice, and if FORM III) CHECKER returns the statement “displays a tree twice” then \mathcal{T} displays a tree twice. \square

6.3.4 Basic cyclic pair algorithm

The algorithm for checking whether or not a basic cyclic pair displays a tree twice is to first get the tree representing the basic cyclic pair. Then send that tree representation to DESIRABLE CONFIG. If we get the statement “needs further checking” then we check if the tree represents a basic⁰ representation. If it does then we send it to BASIC⁰ CHECKER. If it does not then it is of one of three forms, so we check what form it is of then send it to the corresponding form checker algorithm. The algorithm for checking whether or not a basic cyclic pair displays a tree twice is polynomial-time, because all the subroutines of the algorithm run in polynomial-time.

Algorithm : BASIC CHECKER

Input : A tree \mathcal{T} , with root ρ , representing a basic cyclic pair $B-(w, v, P_1, P_2)$. The maximal

subtrees of \mathcal{T} are labeled $\mathcal{T}_1, \mathcal{T}_2$, and the maximal subtrees of \mathcal{T}_1 are labeled $A_{\mathcal{T}_1}, M_{\mathcal{T}_1}$, whilst the maximal subtrees of \mathcal{T}_2 are labeled $A_{\mathcal{T}_2}, M_{\mathcal{T}_2}$. We also have $v^{e_1} \in X_{M_{\mathcal{T}_1}}$ and $v^{e_2} \in X_{M_{\mathcal{T}_2}}$.

Output : The statement “displays a tree twice” or the statement “does not display a tree twice.”

1. Send \mathcal{T} to DESIRABLE CONFIG. If DESIRABLE CONFIG returns “needs further checking” then go to the next step. Otherwise return output of DESIRABLE CONFIG.
2. If \mathcal{T} represents a basic⁰ representation then send \mathcal{T} to BASIC⁰ CHECKER and return output. Otherwise go to the next step.
3. If \mathcal{T} is of form i) then send \mathcal{T} to FORM I) CHECKER and return output. Otherwise go to the next step.
4. If \mathcal{T} is of form ii) then send \mathcal{T} to FORM II) CHECKER and return output. Otherwise go to the next step.
5. Send \mathcal{T} to FORM III) CHECKER and return output.

Theorem 6.3. *Let \mathcal{T} be the representation of the basic cyclic pair $B-(w, v, P_1, P_2)$. Let the root of \mathcal{T} be ρ . Let the maximal subtrees of \mathcal{T} be labeled $\mathcal{T}_1, \mathcal{T}_2$, and let the maximal subtrees of \mathcal{T}_1 be labeled $A_{\mathcal{T}_1}, M_{\mathcal{T}_1}$. Similarly, let the maximal subtrees of \mathcal{T}_2 be labeled $A_{\mathcal{T}_2}, M_{\mathcal{T}_2}$. We also have $v^{e_1} \in X_{M_{\mathcal{T}_1}}$ and $v^{e_2} \in X_{M_{\mathcal{T}_2}}$. If BASIC CHECKER returns the statement “does not display a tree twice” then \mathcal{T} does not display a tree twice, and if BASIC CHECKER returns the statement “displays a tree twice” then \mathcal{T} displays a tree twice. Moreover, BASIC CHECKER runs in $O(|X_{\mathcal{T}}|)$.*

Proof. All the work of this chapter gives the result. □

Chapter 7

An Efficient Algorithm for Deciding whether or not a Phylogenetic Network Displays a Tree Twice

7.1 Introduction

This chapter is the culmination of Chapters 4, 5, and 6. For any given phylogenetic network, we present an efficient algorithm that decides whether or not the network displays a tree twice. The first task is to find any avoidable reticulation vertices in the network. If the number of vertices in the network exceeds a quadratic polynomial in terms of the number of leaves in the network then the network has an avoidable reticulation vertex or a trivial cyclic pair. A trivial cyclic pair is a tree-path cyclic pair that has exactly one two-way tree-path reticulation with respect to its source, namely its sink, and there is no tree-path from the source to a leaf. If a network does not have any avoidable reticulation vertices nor any trivial cyclic pairs then the number of vertices in the network is bounded by a quadratic polynomial in terms of the number of leaves in the network. Therefore the first step in finding avoidable reticulation vertices in a network is to count the number of vertices in the network. Since we assume that the number of leaves in a network is reasonable, if the bound is not exceeded then the number of vertices in the network is reasonable. In order to decide whether or not a network has an avoidable reticulation vertex, for each reticulation vertex in the network we delete it then see if we can still reach every leaf in the network from the root of the network. If we can then the reticulation vertex that we deleted was an avoidable reticulation vertex. If we can not then the reticulation vertex that we deleted was not an avoidable reticulation vertex. If the network has an avoidable reticulation vertex then the network displays a tree twice. If the network

does not have any avoidable reticulation vertices then we start finding cyclic pairs. The number of cyclic pairs in the network can be exponential in terms of the number of reticulation vertices in the network, because for a given reticulation vertex there could be at most 2^{k-1} cyclic pairs that use the given reticulation vertex, where k is the number of reticulation vertices in the network. Fortunately, in such a situation at least $2^{k-1} - 2$ of the cyclic pairs never display a tree twice, so that we only ever have to find and check at most two cyclic pairs for each reticulation vertex in the network. We check a given cyclic pair in the network by sending the tree representation of the associated basic cyclic pair of the processed cyclic pair of the given cyclic pair to BASIC CHECKER. If BASIC CHECKER returns the statement “displays a tree twice” then the network displays a tree twice, and if BASIC CHECKER never returns the statement “displays a tree twice” then the network does not display a tree twice. In this way we get an efficient algorithm that decides whether or not a network displays a tree twice.

7.2 Checking whether or not a Network has any Avoidable Reticulation Vertices

In this section we present an efficient algorithm that decides whether or not a network has any avoidable reticulation vertices.

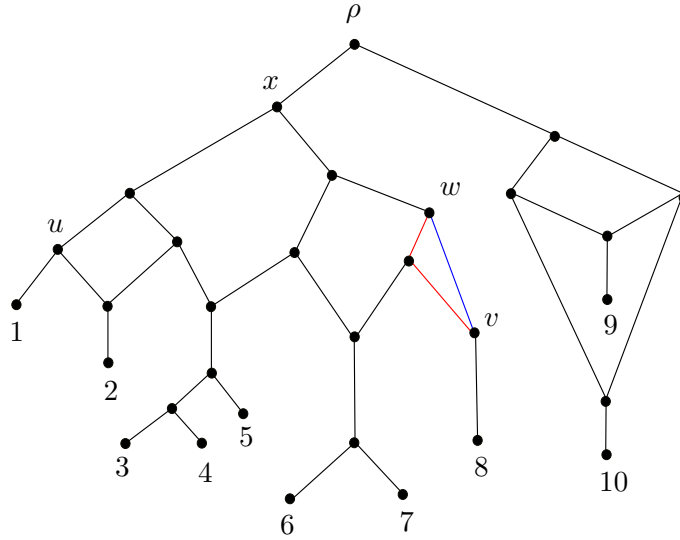
7.2.1 A bound on the number of vertices in a network without any avoidable reticulation vertices nor any trivial cyclic pairs

Here we establish a bound on the number of vertices in a network without any avoidable reticulation vertices nor trivial cyclic pairs. If a reticulation vertex is not avoidable then there exists a leaf such that every path from the root to the leaf passes through the reticulation vertex. There are two types of unavoidable reticulation vertex: The first type has a tree-path to a leaf, which we call a tree-path-to-leaf reticulation vertex, whilst the second type does not have a tree-path to a leaf, which we call a blocked reticulation vertex. The first thing to notice about a blocked reticulation vertex is that there exists a tree-path starting at the blocked reticulation vertex that finishes at the source of a tree-path cyclic pair. In a network without any avoidable reticulation vertices nor any trivial cyclic pairs, the tree-path cyclic pairs whose source is a tree-path descendant of a blocked reticulation have at least two two-way tree-path reticulations with respect to their source, and those two two-way reticulations are either tree-path-to-leaf reticulations and/or blocked reticulations. In this way we can differentiate between different types of blocked reticulation vertices based on the

two-way tree-path reticulation vertices with respect to the source of the tree-path cyclic pair whose source is a tree-path descendant of a given blocked reticulation vertex.

Definition 7.1. Let N be a network on leaf set X . A tree-path cyclic pair $N-(w, v, P_1, P_2)$ is a *trivial cyclic pair* when v is the only two-way tree-path reticulation with respect to w , and there is no tree-path from w to a leaf in X .

Example 7.1. In the following network, the tree-path cyclic pair that has w as its source and v as its sink, whilst one of its main paths is in red and the other is in blue, is a trivial cyclic pair.



Lemma 7.1. Let N be a network without any avoidable reticulation vertices. If there exists a trivial cyclic pair $N-(w, v, P_1, P_2)$ in N then N displays a tree twice.

Proof. Suppose that $N-(w, v, P_1, P_2)$ is a trivial cyclic pair in N . Consider the tree \mathcal{T} representing the associated basic cyclic pair of the processed cyclic pair of $N-(w, v, P_1, P_2)$. We see that \mathcal{T} 's two maximal subtrees are $\mathcal{T}_1 = v^{e_1}$ and $\mathcal{T}_2 = v^{e_2}$; hence $(\mathcal{T}_1)^L = (\mathcal{T}_2)^L$. Thus, by Proposition 5.1, \mathcal{T} displays a tree twice. Therefore the associated basic cyclic pair can not be reduced to a desirable configuration, so, by Corollary 4.2, N displays a tree twice. \square

Definition 7.2. Let N be a network on leaf set X . A vertex u is a *tree-path descendant* of another vertex x if there exists a tree-path from x to u .

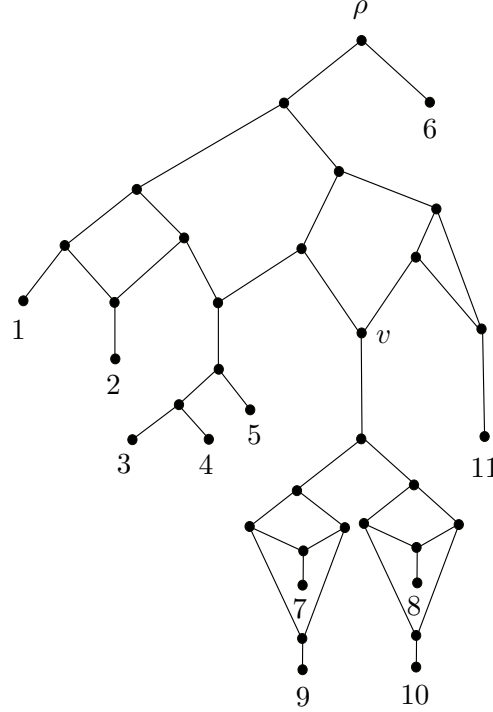
Example 7.2. In Example 7.1, u , w , and v are all tree-path descendants of x .

Definition 7.3. Let N be a network on leaf set X . A reticulation vertex v is called a *tree-path-to-leaf reticulation* when there exists a leaf $\ell \in X$ that is a tree-path descendant of v .

Example 7.3. We see that the reticulation vertex v in Example 7.1 is a tree-path-to-leaf reticulation.

Definition 7.4. Let N be a network on leaf set X . A reticulation vertex v is called a *blocked reticulation* when it is not an avoidable reticulation nor a tree-path-to-leaf reticulation.

Example 7.4. In the following network we see that the reticulation vertex v is a blocked reticulation.

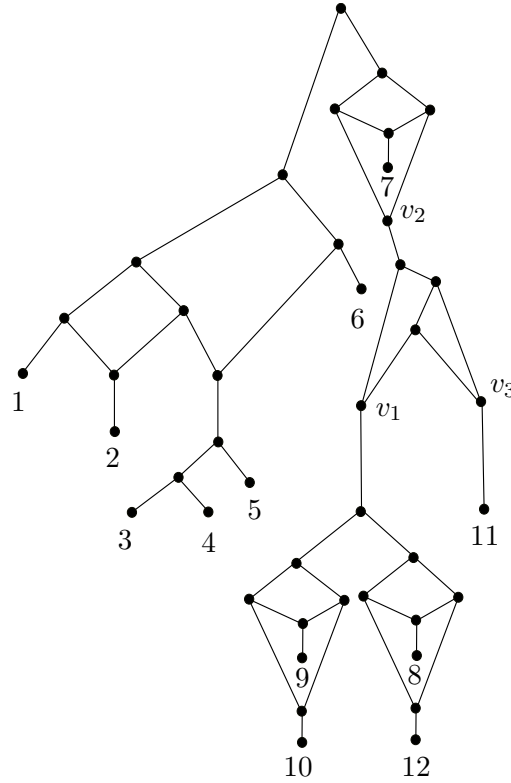


Lemma 7.2. Let N be a network on leaf set X . Let v_1 be a reticulation vertex in N . If v_1 is a blocked reticulation then there exists a tree-path cyclic pair $N-(w, v, P_1, P_2)$ such that w is a tree-path descendant of v_1 .

Proof. Suppose there does not exist a tree-path cyclic pair $N-(w, v, P_1, P_2)$ such that w is a tree-path descendant of v_1 . Suppose v_1 is not a tree-path-to-leaf reticulation; hence every tree-path from v_1 ends at a reticulation vertex. Let \mathcal{D} be the set of all such reticulation vertices. Let \mathcal{P} be the subset of \mathcal{D} such that $v' \in \mathcal{P}$ if and only if every directed path from the root of N to v' passes through v_1 . Suppose $\mathcal{P} \neq \emptyset$ then there exists $v'' \in \mathcal{P}$ such that v'' has no ancestors in \mathcal{P} , and there exists a tree-path cyclic pair $N-(w, v'', P_1, P_2)$ such that w is a tree-path descendant of v_1 - a contradiction. Hence $\mathcal{P} = \emptyset$, so that for every $v' \in \mathcal{D}$ there is a directed path from the root of N to v' that avoids v_1 . Hence for every leaf $\ell \in X$ there is a path from the root of N to ℓ that avoids v_1 . Thus v_1 is an avoidable reticulation vertex. \square

Definition 7.5. Let N be a network on leaf set X . Let v be a blocked reticulation in N . Let Ω be the set of tree-path cyclic pairs $N-(w, v', P_1, P_2)$ such that w is a tree-path descendant of v . We call v a *blocked⁰ reticulation* when for each $N-(w, v', P_1, P_2) \in \Omega$ every two-way reticulation with respect to w is a tree-path-to-leaf reticulation. We call v a *blocked^p reticulation*, where $p \in \mathbb{N}$, when for each $N-(w, v', P_1, P_2) \in \Omega$ every two-way reticulation with respect to w is either a tree-path-to-leaf reticulation or a blocked^g reticulation, where $g \in \mathbb{N} \cup \{0\}$ and $g < p$, and there exists $N-(w_1, v_1, P'_1, P'_2) \in \Omega$ such that v_1 is a blocked^{p-1} reticulation.

Example 7.5. The network shown has v_1 as a blocked⁰ reticulation, v_3 as a tree-path-to-leaf reticulation, and v_2 as a blocked¹ reticulation.



Lemma 7.3. Let N be a network on leaf set X such that N has no avoidable reticulations nor any trivial cyclic pairs. Let the set of all blocked reticulations in N be called B . Let the set of all blocked^p reticulations in N , where $p \in \mathbb{N} \cup \{0\}$, be called B_p . If $|B| = n$ for some $n \in \mathbb{N}$ then $|B| = |B_q| + |B_{q-1}| + \dots + |B_1| + |B_0|$, for some $q \leq n-1$. Moreover, $|B_g| \geq 1$ for all $g \in \{0, 1, \dots, q\}$, and for all $t \in \mathbb{N}$ such that $t > q$ we have $|B_t| = 0$.

Proof. Suppose $|B| = n$ for some $n \in \mathbb{N}$. We claim that there exists $k \leq n-1$ such that $|B_k| \geq 1$ and $|B_{k+1}| = 0$. Consider $q = n-1$. If $|B_q| = 0$ then it follows from the definition of a blocked^{q+1} reticulation that $|B_{q+1}| = 0$; in fact, it follows that $|B_t| = 0$ for all $t \in \mathbb{N}$ such that $t > q$. Hence

we can show that there exists $k \leq q - 1$ such that $|B_k| \geq 1$ and $|B_{k+1}| = 0$. Suppose that $|B_q| \geq 1$ and suppose for contradiction that $|B_{q+1}| \geq 1$. By the definition of a blocked^q reticulation and $|B_q| \geq 1$, we have $|B_{q-1}| \geq 1$; similarly, $|B_{q-2}| \geq 1$, and it follows that for all $g \in \{0, 1, \dots, q\}$ we have $|B_g| \geq 1$. Again by the definition of a blocked^q reticulation, we see that $B_q \cap B_{q-1} = \emptyset$; it follows that $B_{q+1} \cap B_q \cap \dots \cap B_1 \cap B_0 = \emptyset$. Therefore $|B_{q+1}| + |B_q| + \dots + |B_1| + |B_0| > n$. It is the case that $|B| \geq |B_{q+1}| + |B_q| + \dots + |B_1| + |B_0|$; hence $|B| > n$ - a contradiction. Thus there exists $k \leq n - 1$ such that $|B_k| \geq 1$ and $|B_{k+1}| = 0$. All in all we get $|B| = |B_q| + |B_{q-1}| + \dots + |B_1| + |B_0|$, for some $q \leq n - 1$. Moreover, $|B_g| \geq 1$ for all $g \in \{0, 1, \dots, q\}$, and for all $t \in \mathbb{N}$ such that $t > q$ we have $|B_t| = 0$. \square

Lemma 7.4. *Let N be a network on leaf set X such that N has no avoidable reticulations nor any trivial cyclic pairs. Let the set of all blocked reticulations in N be called B . Let the set of all blocked^p reticulations in N , where $p \in \mathbb{N} \cup \{0\}$, be called B_p . Let T_r be the set of tree-path-to-leaf reticulations in N . Then $|T_r| \leq |X|$, $|B_0| \leq |T_r|$, and $|B_q| \leq |B_{q-1}|$ for all $q \in \mathbb{N}$.*

Proof. We want to find a one to one function from the set of tree-path-to-leaf reticulations in N to the set of leaves in N . Let v be any tree-path-to-leaf reticulation in N then there exists a leaf $\ell_v \in X$ such that ℓ_v is a tree-path descendant of v . We see that no other tree-path-to-leaf reticulation in N has ℓ_v as a tree-path descendant, so define the function $f : T_r \rightarrow X$ as $f(v) = \ell_v$. Hence f is a one to one function. Similarly, there exists a one to one function from the set of blocked⁰ reticulations to the set of tree-path-to-leaf reticulations. Let v_1 be any blocked⁰ reticulation in N then there exists a tree-path cyclic pair $N-(w, v', P_1, P_2)$ such that w is a tree-path descendant of v_1 . We see that v' is a tree-path-to-leaf reticulation and every directed path from the root of N to v' passes through w . Since $N-(w, v', P_1, P_2)$ is a tree-path cyclic pair, no blocked⁰ reticulation in N , except for v_1 , has v' as a tree-path descendant, so define $f_1 : B_0 \rightarrow T_r$ as $f_1(v_1) = v'$. Hence f_1 is a one to one function. Likewise, there exists a one to one function from the set of blocked^q reticulations to the set of blocked^{q-1} reticulations for all $q \in \mathbb{N}$. Therefore $|T_r| \leq |X|$, $|B_0| \leq |T_r|$, and $|B_q| \leq |B_{q-1}|$ for all $q \in \mathbb{N}$. \square

Proposition 7.1. *Let N be a network on leaf set X such that N has no avoidable reticulations nor any trivial cyclic pairs. Let the set of all blocked reticulations in N be called B . Let the set of all blocked^p reticulations in N , where $p \in \mathbb{N} \cup \{0\}$, be called B_p . Let T_r be the set of tree-path-to-leaf reticulations in N . Then $|B| \leq (q + 1)|X|$ for some $q \in \mathbb{N} \cup \{0\}$, and $(q + 2) \leq |X|$.*

Proof. By Lemma 7.3, $|B| = |B_q| + |B_{q-1}| + \dots + |B_1| + |B_0|$, for some $q \in \mathbb{N} \cup \{0\}$. By Lemma 7.4, it follows that $|B_g| \leq |X|$ for all $g \leq q$, where $g \in \mathbb{N} \cup \{0\}$. Hence $|B| \leq (q + 1)|X|$. Since

N has no trivial cyclic pairs, for each tree-path cyclic pair $N-(w, v_1, P_1, P_2)$ there are at least two two-way reticulations with respect to w , or there is exactly one two-way reticulation with respect to w and there is a tree-path from w to a leaf in X . Consider any blocked ^{q} reticulation v in N . Then there exists a tree-path cyclic pair $N-(w, v_1, P_1, P_2)$ such that w is a tree-path descendant of v and v_1 is a blocked ^{$q-1$} reticulation. There exists another two-way reticulation with respect to w , so let us assume that that two-way reticulation is a tree-path-to-leaf reticulation. Now consider v_1 - it follows that there exists a tree-path cyclic pair $N-(w_1, v_2, P_1^1, P_2^1)$ such that w_1 is a tree-path descendant of v_1 and v_2 is a blocked ^{$q-2$} reticulation (assuming that $q \geq 2$). There exists another two-way reticulation with respect to w_1 , so let us assume that that two-way reticulation is a tree-path-to-leaf reticulation. We can continue like this until we reach a blocked⁰ reticulation v_q , and there exists a tree-path cyclic pair $N-(w_q, v_{q+1}, P_1^q, P_2^q)$ such that w_q is a tree-path descendant of v_q and v_{q+1} is a tree-path-to-leaf reticulation. There exists another reticulation with respect to w_q , and that two-way reticulation is a tree-path-to-leaf reticulation. Hence there are at least $(q+2)$ tree-path-to-leaf reticulations v' that are descendants of v such that every directed path from the root of N to v' passes through v . Therefore $(q+2)|B_q| \leq |T_r|$, and, by Lemma 7.4, $(q+2)|B_q| \leq |X|$. Now, by Lemma 7.3, $|B_q| \geq 1$, so $(q+2) \leq \frac{|X|}{|B_q|}$ and $\frac{|X|}{|B_q|} \leq |X|$; thus $(q+2) \leq |X|$. \square

Theorem 7.1. *Let N be a network on leaf set X such that N has no avoidable reticulations nor any trivial cyclic pairs. Let the set of all blocked reticulations in N be called B . Let the set of all blocked ^{p} reticulations in N , where $p \in \mathbb{N} \cup \{0\}$, be called B_p . Let T_r be the set of tree-path-to-leaf reticulations in N . Then $|V| \leq 2(|X|^2 + |X|) - 1$, where V is the set of vertices in N .*

Proof. Since N has no avoidable reticulations, $k = |B| + |T_r|$, where k is the number of reticulation vertices in N . By Proposition 7.1 and Lemma 7.4, we have $|B| + |T_r| \leq (q+1)|X| + |X|$. Thus $k \leq (q+2)|X|$. By Proposition 7.1, $(q+2)|X| \leq |X|^2$. Hence $k \leq |X|^2$. By [26, Equation 5], we have $|X| + k = \frac{|V|+1}{2}$, so $\frac{|V|+1}{2} \leq |X|^2 + |X|$. Therefore $|V| \leq 2(|X|^2 + |X|) - 1$. \square

7.2.2 An algorithm for deciding whether or not a network has an avoidable reticulation vertex

It is simple to decide whether or not a network has an avoidable reticulation vertex. For any given reticulation vertex in the network, delete the given reticulation vertex and then see if there is a directed path from the root of the network to any given leaf in the leaf set of the network. If for every leaf in the leaf set of the network there still remains a directed path from the root of the network to it then the given reticulation vertex is an avoidable reticulation vertex. If there exists a leaf in the leaf set of the network such that there is no longer a directed path from the root of the

network to it then the given reticulation vertex is not an avoidable reticulation vertex. In this way we can decide whether or not a network has an avoidable reticulation vertex.

Algorithm: AVOIDABLE FINDER

Input: A network N on leaf set X with root ρ .

Output: The statement “has an avoidable reticulation vertex” or the statement “does not have an avoidable reticulation vertex.”

1. Make a list of reticulation vertices in N , call them v_1, v_2, \dots, v_k , where $k \in \mathbb{N}$.
2. If the list from the previous step is empty then go to Step 7. Take the first reticulation vertex on the list, say v_i where $i \in \{1, 2, \dots, k\}$.
3. Delete v_i in N , and, for each leaf $\ell \in X$, see if there is a directed path from ρ to ℓ .
4. If there exists a leaf $\ell \in X$ such that there is no longer a directed path from ρ to ℓ , after v_i is deleted in N , then delete v_i from the list of reticulation vertices in N and return to Step 2.
5. If for each leaf $\ell \in X$ there is still a directed path from ρ to ℓ even after v_i is deleted in N then go to Step 6.
6. Return the statement “has an avoidable reticulation vertex.”
7. Return the statement “does not have an avoidable reticulation vertex.”

Lemma 7.5. *Let N be a network on leaf set X with root ρ . If AVOIDABLE FINDER returns the statement “has an avoidable reticulation vertex” then N has an avoidable reticulation vertex, and if AVOIDABLE FINDER returns the statement “does not have an avoidable reticulation vertex” then N does not have an avoidable reticulation vertex.*

Proof. Suppose AVOIDABLE FINDER returns the statement “has an avoidable reticulation vertex” then there exists a reticulation vertex v in N such that for each leaf $\ell \in X$ there is a directed path from ρ to ℓ after v is deleted in N . This means that for every leaf $\ell \in X$ there is a directed path from ρ to ℓ that avoids v . Therefore v is an avoidable reticulation vertex in N . Suppose AVOIDABLE FINDER returns the statement “does not have an avoidable reticulation vertex” then let v' be any reticulation vertex in N . When v' is deleted in N there exists a leaf $\ell' \in X$ such that there is no directed path from ρ to ℓ' . This means that every directed path in N from ρ to ℓ' passes through v' . Therefore every reticulation vertex in N is not an avoidable reticulation vertex. \square

7.3 Finding Cyclic Pairs that Need Checking

Here we present an algorithm for finding cyclic pairs that need checking, in a network without any avoidable reticulation vertices. The cyclic pairs that do not need to be checked are the cyclic pairs with three or more reticulation vertices on their main paths. These cyclic pairs do not need to be checked because they can be reduced to desirable configurations, and therefore they do not display a tree twice.

Algorithm: CYCLIC PAIR FINDER

Input: A network N on leaf set X without any avoidable reticulation vertices.

Output: A list of cyclic pairs.

1. Make a list of reticulation vertices in N , call them v_1, v_2, \dots, v_k , where $k \in \mathbb{N}$.
2. If the list from the previous step is empty then go to Step 14. Otherwise take the first element v_i , where $i \in \{1, 2, \dots, k\}$, on the list of reticulation vertices.
3. Find a cyclic pair that involves v_i , and let it be called $N-(w_i^1, v_i, P_1^i, P_2^i)$. Let the number of reticulation vertices on P_1^i be $K_{P_1^i}$, and let the number of reticulation vertices on P_2^i be $K_{P_2^i}$.
4. If $K_{P_1^i} \geq 2$ and $K_{P_2^i} \geq 2$ then go to Step 11 followed by Step 12.
5. If $K_{P_1^i} > 2$ and $K_{P_2^i} = 1$ then go to Step 11 followed by Step 13.
6. If $K_{P_1^i} = 1$ and $K_{P_2^i} > 2$ then go to Step 12.
7. If $K_{P_1^i} = 2$ and $K_{P_2^i} = 1$ then go to Step 10 followed by Step 11 then go to Step 13.
8. If $K_{P_1^i} = 1$ and $K_{P_2^i} = 2$ then go to Step 10 followed by Step 12.
9. If $K_{P_1^i} = 1$ and $K_{P_2^i} = 1$ then go to Step 10 followed by Step 13.
10. Add $N-(w_i^1, v_i, P_1^i, P_2^i)$ to the list of cyclic pairs.
11. Let v_i^1 be the reticulation vertex on P_1^i such that there exists a tree-path from v_i^1 to v_i . Let $e_1^{v_i^1}$ be the reticulation edge of v_i^1 that is not on P_1^i . Find a cyclic pair $N-(w_i^2, v_i, \hat{P}_1^i, \hat{P}_2^i)$ such that \hat{P}_1^i has the edge $e_1^{v_i^1}$. Let the number of reticulation vertices on \hat{P}_1^i be $K_{\hat{P}_1^i}$, and let the number of reticulation vertices on \hat{P}_2^i be $K_{\hat{P}_2^i}$. If $K_{\hat{P}_1^i} = 2$ and $K_{\hat{P}_2^i} = 1$ then add $N-(w_i^2, v_i, \hat{P}_1^i, \hat{P}_2^i)$ to the list of cyclic pairs. If $K_{\hat{P}_1^i} \geq 2$ and $K_{\hat{P}_2^i} \geq 2$; or $K_{\hat{P}_1^i} > 2$; or $K_{\hat{P}_2^i} > 2$ then do not add $N-(w_i^2, v_i, \hat{P}_1^i, \hat{P}_2^i)$ to the list of cyclic pairs.

12. Let v_i^2 be the reticulation vertex on P_2^i such that there exists a tree-path from v_i^2 to v_i . Let $e_1^{v_i^2}$ be the reticulation edge of v_i^2 that is not on P_2^i . Find a cyclic pair $N-(w_i^2, v_i, \tilde{P}_1^i, \tilde{P}_2^i)$ such that \tilde{P}_2^i has the edge $e_1^{v_i^2}$. Let the number of reticulation vertices on \tilde{P}_2^i be $K_{\tilde{P}_2^i}$, and let the number of reticulation vertices on \tilde{P}_1^i be $K_{\tilde{P}_1^i}$. If $K_{\tilde{P}_2^i} = 2$ and $K_{\tilde{P}_1^i} = 1$ then add $N-(w_i^2, v_i, \tilde{P}_1^i, \tilde{P}_2^i)$ to the list of cyclic pairs and go to Step 13. If $K_{\tilde{P}_2^i} \geq 2$ and $K_{\tilde{P}_1^i} \geq 2$; or $K_{\tilde{P}_2^i} > 2$; or $K_{\tilde{P}_1^i} > 2$ then go to Step 13.
13. Delete v_i from the list of reticulation vertices and return to Step 2.
14. Return the list of cyclic pairs.

Lemma 7.6. *Let N be a network on leaf set X without any avoidable reticulation vertices. Let L be the list of cyclic pairs outputted by CYCLIC PAIR FINDER, upon input of N . Let $N-(w, v, P_1, P_2)$ be a cyclic pair of N . Let K_{P_1} be the number of reticulation vertices on P_1 , and let K_{P_2} be the number of reticulation vertices on P_2 . If $N-(w, v, P_1, P_2)$ is not on L then either $K_{P_1} \geq 2$ and $K_{P_2} \geq 2$; or $K_{P_1} > 2$ and $K_{P_2} = 1$; or $K_{P_1} = 1$ and $K_{P_2} > 2$.*

Proof. (Proof by contrapositive.) Suppose that $K_{P_1} = 1$ and $K_{P_2} = 1$ then there is only one cyclic pair involving v , and we would have found it in Step 3 of CYCLIC PAIR FINDER and subsequently added it to L ; hence $N-(w, v, P_1, P_2)$ is on L . Without loss of generality, suppose that $K_{P_1} = 2$ and $K_{P_2} = 1$. Let v^* be the reticulation vertex on P_1 such that v is a tree-path descendant of v^* . Let $N-(w', v, P'_1, P'_2)$ be the cyclic pair found in Step 3 of CYCLIC PAIR FINDER. Let $K_{P'_1}$ be the number of reticulation vertices on P'_1 , and let $K_{P'_2}$ be the number of reticulation vertices on P'_2 . Without loss of generality, suppose that v^* is on P'_1 . If $K_{P'_2} = 1$ then we must have $P_2 = P'_2$. In Step 11 of CYCLIC PAIR FINDER we would have found $N-(w, v, P_1, P_2)$ and added it to L , so suppose that $K_{P'_2} \geq 2$. Let v_2^* be the reticulation vertex on P'_2 such that v is a tree-path descendant of v_2^* . Then w is a tree-path descendant of v_2^* . In Step 11 of CYCLIC PAIR FINDER we would have found $N-(w, v, P_1, P_2)$ and added it to L . Therefore $N-(w, v, P_1, P_2)$ is on L . \square

Proposition 7.2. *Let N be a network without any avoidable reticulation vertices. Let $N-(w, v, P_1, P_2)$ be a cyclic pair of N . Let L be the list of cyclic pairs that is returned when CYCLIC PAIR FINDER is applied to N . If $N-(w, v, P_1, P_2)$ is not on L then the associated basic cyclic pair of the processed cyclic pair of $N-(w, v, P_1, P_2)$ can be reduced to a desirable configuration.*

Proof. Suppose $N-(w, v, P_1, P_2)$ is not on L then there must be too many reticulation vertices on P_1 and/or P_2 . Let K_{P_1} be the number of reticulation vertices on P_1 , and let K_{P_2} be the number of reticulation vertices on P_2 . Then, by Lemma 7.6, either $K_{P_1} \geq 2$ and $K_{P_2} \geq 2$; or

$K_{P_1} > 2$ and $K_{P_2} = 1$; or $K_{P_1} = 1$ and $K_{P_2} > 2$. Without loss of generality, suppose that $K_{P_1} > 2$ and $K_{P_2} = 1$ then there exists a reticulation vertex v_1 on P_1 such that $v_1 \neq v$ and v is a tree-path descendant of v_1 , and there exists another reticulation vertex v_2 on P_1 such that $v_2 \neq v_1$ and v_1 is a tree-path descendant of v_2 . Since N has no avoidable reticulation vertices, v_1 is either a tree-path-to-leaf reticulation vertex or a blocked reticulation vertex; likewise, v_2 is either a tree-path-to-leaf reticulation vertex or a blocked reticulation vertex. Without loss of generality, suppose that both v_1 and v_2 are blocked reticulation vertices then there exists a tree-path cyclic pair $N-(w', v_3, P'_1, P'_2)$ such that w' is a tree-path descendant of v_1 , and there exists a tree-path cyclic pair $N-(w'', v_4, P''_1, P''_2)$ such that w'' is a tree-path descendant of v_2 . Now consider the associated basic cyclic pair $B-(w, v, P_1^*, P_2^*)$ of the processed cyclic pair of $N-(w, v, P_1, P_2)$. Consider the tree representation \mathcal{T} of $B-(w, v, P_1^*, P_2^*)$. Let Γ be the reduction of \mathcal{T} where $\Gamma = \{v^{e_1}, v^{e_2}, v_3^{e_1}, v_3^{e_2}, v_4^{e_1}, v_4^{e_2}\}$. We see that $\mathcal{T}|\Gamma$ never displays a tree twice, by Proposition 6.3, as one child of the root of $\mathcal{T}|\Gamma$ has both its children being ancestors of partners. Therefore the associated basic cyclic pair of the processed cyclic pair of $N-(w, v, P_1, P_2)$ can be reduced to a desirable configuration. The case when $K_{P_1} \geq 2$ and $K_{P_2} \geq 2$ is similar. \square

7.4 The Efficient Algorithm

We now have all the tools that we need in order to create an efficient algorithm for deciding whether or not a phylogenetic network displays a tree twice. Given a phylogenetic network, the first step is to count the number of vertices in the network. If the number of vertices in the network exceeds $2(|X|^2 + |X|) - 1$, where $|X|$ is the number of leaves in the network, then the network has an avoidable reticulation vertex or a trivial cyclic pair; in both cases the network displays a tree twice. If the number of vertices in the network does not exceed $2(|X|^2 + |X|) - 1$ then we run the network through AVOIDABLE FINDER. If the statement “has an avoidable reticulation vertex” is outputted then the network displays a tree twice. If the statement “does not have an avoidable reticulation vertex” is outputted then the network does not have any avoidable reticulation vertices, and in that case we run the network through CYCLIC PAIR FINDER. For each cyclic pair on the list of cyclic pairs outputted by CYCLIC PAIR FINDER, we get the tree representation of the associated basic cyclic pair of the processed cyclic pair of that cyclic pair, and we send that tree representation to BASIC CHECKER. If BASIC CHECKER returns the statement “displays a tree twice” then the network displays a tree twice. If BASIC CHECKER returns the statement “does not display a tree twice” for each cyclic pair on the list of cyclic pairs then the network does not display a tree twice. Since we assume that the number of leaves in the network is reasonable,

the algorithm for deciding whether or not a phylogenetic network displays a tree twice is efficient.

Algorithm: DISPLAYS TWICE

Input: A phylogenetic network N on leaf set X .

Output: The statement “displays a tree twice” or the statement “does not display a tree twice.”

1. Start counting the number of vertices in N . If the count exceeds $2(|X|^2 + |X|) - 1$ then go to Step 5.
2. Send N to AVOIDABLE FINDER. If the statement “has an avoidable reticulation vertex” is returned then go to Step 5.
3. Send N to CYCLIC PAIR FINDER, and let the list of cyclic pairs that is outputted be called L . For each cyclic pair $N-(w, v, P_1, P_2)$ on L send the tree representation \mathcal{T} of the associated basic cyclic pair of the processed cyclic pair of $N-(w, v, P_1, P_2)$ to BASIC CHECKER. If the statement “displays a tree twice” is returned then go to Step 5.
4. Return the statement “does not display a tree twice.”
5. Return the statement “displays a tree twice.”

Theorem 7.2. *Let N be a phylogenetic network on leaf set X . If DISPLAYS TWICE returns the statement “displays a tree twice,” upon input of N , then N displays a tree twice, and if DISPLAYS TWICE returns the statement “does not display a tree twice,” upon input of N , then N does not display a tree twice.*

Proof. Suppose DISPLAYS TWICE returns the statement “displays a tree twice,” upon input of N , then either the number of vertices in N exceeds $2(|X|^2 + |X|) - 1$; or AVOIDABLE FINDER returns the statement “has an avoidable reticulation vertex”; or BASIC CHECKER returns the statement “displays a tree twice,” upon input of the tree representation of the associated basic cyclic pair of the processed cyclic pair of one of the cyclic pairs on the list of cyclic pairs outputted by CYCLIC PAIR FINDER. If the number of vertices exceeds $2(|X|^2 + |X|) - 1$ then, by Theorem 7.1, N has an avoidable reticulation vertex or a trivial cyclic pair. If N has an avoidable reticulation vertex then, by Lemma 1.3, N displays a tree twice, and if N has a trivial cyclic pair then, by Lemma 7.1, N displays a tree twice. If AVOIDABLE FINDER returns the statement “has an avoidable reticulation vertex,” upon input of N , then, by Lemma 7.5, N has an avoidable reticulation vertex; hence N displays a tree twice. If BASIC CHECKER returns the statement “displays a tree twice,” upon

input of the tree representation \mathcal{T} of the associated basic cyclic pair of the processed cyclic pair of one of the cyclic pairs on the list of cyclic pairs outputted by CYCLIC PAIR FINDER, then, by Theorem 6.3, \mathcal{T} displays a tree twice. Thus \mathcal{T} 's associated basic cyclic pair can not be reduced to a desirable configuration. Therefore, by Corollary 4.2, N displays a tree twice. Suppose DISPLAYS TWICE returns the statement “does not display a tree twice,” upon input of N , then the number of vertices in N does not exceed $2(|X|^2 + |X|) - 1$, and AVOIDABLE FINDER returns the statement “does not have an avoidable reticulation vertex,” upon input of N , which implies that N does not have an avoidable reticulation vertex, by Lemma 7.5. Therefore BASIC CHECKER returns “does not display a tree twice,” upon input of the tree representation of the associated basic cyclic pair of the processed cyclic pair of any cyclic pair on the list of cyclic pairs outputted by CYCLIC PAIR FINDER, upon input of N , which implies that the associated basic cyclic pair of the processed cyclic pair of any cyclic pair on the list of cyclic pairs can be reduced to a desirable configuration. By Proposition 7.2, the associated basic cyclic pair of the processed cyclic pair of any cyclic pair not on the list of cyclic pairs outputted by CYCLIC PAIR FINDER can be reduced to a desirable configuration. Therefore the associated basic cyclic pair of the processed cyclic pair of any cyclic pair in N can be reduced to a desirable configuration; thus, by Corollary 4.2, N does not display a tree twice. \square

Theorem 7.3. *Let N be a phylogenetic network on leaf set X , and let V be the set of vertices of N . DISPLAYS TWICE runs in quartic-time in terms of $|X|$.*

Proof. DISPLAYS TWICE stops if the number of vertices in N exceeds $2(|X|^2 + |X|) - 1$, so the maximum number of steps DISPLAYS TWICE can take in its first step is $2(|X|^2 + |X|) - 1$. The second step of DISPLAYS TWICE employs the algorithm AVOIDABLE FINDER, and the maximum number of steps AVOIDABLE FINDER can take is k , where k is the number of reticulation vertices in N . Provided that we get to the second step, the maximum number of steps that the second step can take is less than or equal to $2(|X|^2 + |X|) - 1$, as $k \leq |V|$ and $|V| \leq 2(|X|^2 + |X|) - 1$. The third step of DISPLAYS TWICE employs CYCLIC PAIR FINDER and BASIC CHECKER. The running time of BASIC CHECKER is of $O(|X_{\mathcal{T}}|)$, where \mathcal{T} is the tree representation of a basic cyclic pair. Now $|X_{\mathcal{T}}| \leq 2|V|$, where V is the vertex set of N . Since $|V| \leq 2(|X|^2 + |X|) - 1$, we see that, in this case, BASIC CHECKER runs in $O(|X|^2)$. Since the number of cyclic pairs that CYCLIC PAIR FINDER can find is less than or equal to two times the number of vertices in N and $|V| \leq 2(|X|^2 + |X|) - 1$, we see that DISPLAYS TWICE runs in $O(|X|^4)$. \square

Chapter 8

A Class of Locally Independent Networks

8.1 Introduction

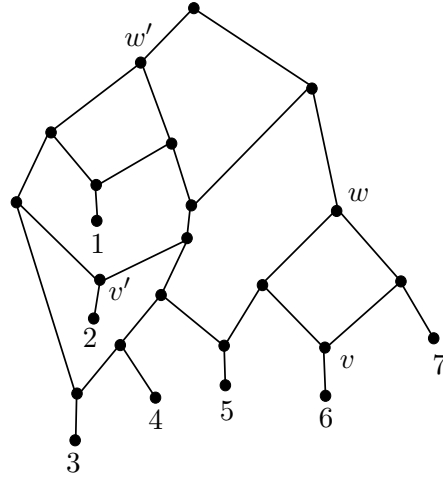
This chapter is about a class of phylogenetic networks for which there exists a quadratic-time algorithm for counting the number of distinct trees displayed by a network in the class. A network that is in that class is called a locally independent network. It is shown in this chapter that a locally independent network has no avoidable reticulation vertices, no blocked reticulation vertices, and no trivial cyclic pairs. It follows that the size of the leaf set of a locally independent network bounds the number of vertices in the network. This allows one to count the number of distinct trees displayed by a locally independent network in quadratic-time in terms of the leaf set of the network.

8.2 Definition and Properties of a Locally Independent Network

Here we define what a locally independent network is, as well as showing that a locally independent network has no avoidable reticulation vertices. This leads to a bound on the number of vertices of a locally independent network in terms of the leaf set of the network.

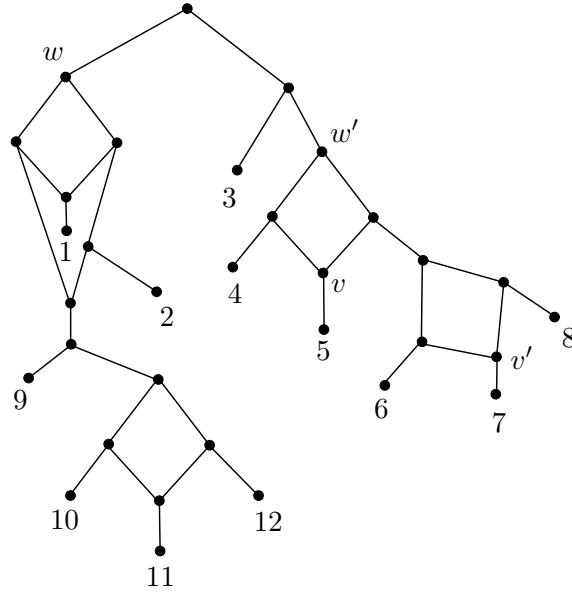
Definition 8.1. A cyclic pair $N-(w, v, P_1, P_2)$ has the *tree-path to a leaf property* when there exists a tree-path from w to a leaf of N .

Example 8.1. In the following figure, the cyclic pair that has w as its source and v as its sink has the tree-path to a leaf property. The cyclic pair that has w' as its source and v' as its sink does not have the tree-path to a leaf property.



Definition 8.2. Let N be a network such that every cyclic pair $N-(w, v, P_1, P_2)$ of N is a maximal tree-path cyclic pair with the tree-path to a leaf property. Then N is said to be a *locally independent network*.

Example 8.2. The following network is an example of a locally independent network.



Definition 8.3. Let N be a network on leaf set X . Let $N-(w, v, P_1, P_2)$ be a cyclic pair of N . Let v' be a two-way tree-path reticulation with respect to w . We say that v' is a *main two-way tree-path reticulation with respect to w* when the two tree-paths from w to v' are edge-disjoint.

Example 8.3. In the network from Example 8.2 we see that the reticulation vertex labeled v is a main two-way tree-path reticulation with respect to w' , whilst the reticulation vertex labeled v' is a two-way tree-path reticulation with respect to w' but not a main two-way tree-path reticulation with respect to w' .

Lemma 8.1. *Let N be a network. If N is a locally independent network then N has no avoidable reticulation vertices.*

Proof. Suppose that N has an avoidable reticulation vertex. Let v be an avoidable reticulation vertex of N such that there are no descendants of v that are themselves avoidable reticulation vertices. Hence every tree-path starting at v ends at an unavoidable reticulation vertex. Let the set of all such unavoidable reticulation vertices be called D . Since v is an avoidable reticulation vertex, there exists a reticulation vertex $v' \in D$ such that v' is not a two-way tree-path reticulation with respect to v . There is no way that v' can form a maximal tree-path cyclic pair. Therefore N is not a locally independent network. \square

Lemma 8.2. *Let N be a locally independent network on leaf set X . Let V be the vertex set of N then $|V| < 4|X|$.*

Proof. Let k be the number of reticulation vertices in N . By Lemma 8.1, N has no avoidable reticulation vertices, and, since every cyclic pair in N has the tree-path to a leaf property, there are no blocked reticulation vertices nor any trivial cyclic pairs in N . Therefore, by Lemma 7.4, $k \leq |X|$. By [26, Equation 5], $|X| + k = \frac{|V|+1}{2}$. Thus $2(|X| + k) - 1 = |V|$, and $|V| \leq 2(2|X|) - 1$, because $k \leq |X|$. Therefore $|V| < 4|X|$. \square

8.3 An Equivalence Relation

Let $N-(w, v, P_1, P_2)$ be any cyclic pair in a locally independent network N . By definition, $N-(w, v, P_1, P_2)$ is a maximal tree-path cyclic pair with the tree-path to a leaf property. Hence we can immediately get the tree $\mathcal{T}^{(w, v, P_1, P_2)}$ representing the associated basic cyclic pair of $N-(w, v, P_1, P_2)$. Let $N-(w, v', P'_1, P'_2)$ be any other cyclic pair that shares the same source as $N-(w, v, P_1, P_2)$, namely w . We see that the tree $\mathcal{T}^{(w, v', P'_1, P'_2)}$ representing the associated basic cyclic pair of $N-(w, v', P'_1, P'_2)$ is isomorphic to $\mathcal{T}^{(w, v, P_1, P_2)}$. Thus we can define an equivalence relation on the set of cyclic pairs in N .

Definition 8.4. Let $N-(w, v, P_1, P_2)$ and $N-(w', v', P'_1, P'_2)$ be two cyclic pair in a locally independent network N . We say that $N-(w, v, P_1, P_2)$ and $N-(w', v', P'_1, P'_2)$ are related when $w = w'$. We call this relation the *locally independent relation on N* .

Example 8.4. In the network from Example 8.2, the two cyclic pairs that have w as their source are related under the locally independent relation.

Lemma 8.3. *Let N be a locally independent network. The locally independent relation on N is an equivalence relation.*

Proof. It is easy to see that the locally independent relation on N is reflexive, symmetric, and transitive. \square

8.4 Algorithm

The locally independent relation on N partitions the set of cyclic pairs in N into equivalence classes. Let J be any such equivalence class of N . Let $N-(w, v, P_1, P_2)$ be any representative of J . We now need to count the number of trees displayed by the associated basic cyclic pair of $N-(w, v, P_1, P_2)$. Let k be the number of main two-way tree-path reticulation vertices with respect to w in the associated basic cyclic pair of $N-(w, v, P_1, P_2)$. It is the case that the number of networks/trees displayed by the associated basic cyclic pair of $N-(w, v, P_1, P_2)$, via general switchings with the following property: for each reticulation vertex v' that is not a main two-way tree-path reticulation with respect to w both reticulation edges of v' are in the general switchings, is either 2^k or $2^k - 1$. This is the case because $N-(w, v, P_1, P_2)$ is a maximal tree-path cyclic pair with the tree-path to a leaf property. That fact together with the fact that a representative cyclic pair of one equivalence class is completely independent of a representative cyclic pair of another equivalence class leads to the following algorithm.

Algorithm : COUNTINGLOCAL

Input : A locally independent network N together with an index j .

Output : The number of distinct trees displayed by N .

1. Compile the list of all sources of cyclic pairs in N , call them w_1, w_2, \dots, w_n . If there are no cyclic pairs in N then return the number one.
2. Set the index j to be one.
3. If $j = n + 1$ then multiply all the numbers together on the list from Step 1 and return output.
4. For w_j find a cyclic pair $N-(w_j, v_j, P_1^j, P_2^j)$, and get the tree \mathcal{T}_j representing $N-(w_j, v_j, P_1^j, P_2^j)$, where \mathcal{T}_j has maximal subtrees $\mathcal{T}_1^j, \mathcal{T}_2^j$ and \mathcal{T}_1^j has maximal subtrees $A_{\mathcal{T}_1^j}, M_{\mathcal{T}_1^j}$.

5. Relabel the maximal subtrees of \mathcal{T}_j such that $X_{\mathcal{T}_1^j}$ contains a fixed leaf, and relabel the maximal subtrees of \mathcal{T}_1^j such that $X_{A_{\mathcal{T}_1^j}}$ contains a fixed leaf.
6. If $X_{\mathcal{T}_2^j}$ or $X_{M_{\mathcal{T}_1^j}}$ contains a fixed leaf then go to Step 8.
7. If $(\mathcal{T}_2^j)^L \mid (X_{M_{\mathcal{T}_1^j}})^L = (M_{\mathcal{T}_1^j})^L$ then on the list from Step 1 replace w_j with the number $2^{k_j} - 1$, where k_j is the number of main two-way tree-path reticulations with respect to w_j , and return to Step 3 setting j to be $j + 1$. Otherwise go to Step 8.
8. On the list from step 1 replace w_j with the number 2^{k_j} and return to Step 3 setting j to be $j + 1$.

Theorem 8.1. *Let N be a locally independent network. COUNTINGLOCAL returns the number of distinct trees displayed by N upon input of N . Moreover, COUNTINGLOCAL runs in polynomial-time.*

Proof. If N is a tree then COUNTINGLOCAL will return the number one at Step 1. Hence COUNTINGLOCAL returns the number of distinct trees displayed by N when N is a tree. Suppose N is not a tree then N has cyclic pairs. Let all the sources of cyclic pairs in N be compiled into a list, say w_1, w_2, \dots, w_n . Let w_j be any such source in N , so $j \in \{1, 2, \dots, n\}$. Let $N-(w_j, v_j, P_1^j, P_2^j)$ be a cyclic pair that has w_j as its source. We see that $N-(w_j, v_j, P_1^j, P_2^j)$ is a representative of an equivalence class under the locally independent relation. Let \mathcal{T}_j be the tree representing the associated basic cyclic pair of $N-(w_j, v_j, P_1^j, P_2^j)$. Let the maximal subtrees of \mathcal{T}_j be $\mathcal{T}_1^j, \mathcal{T}_2^j$, and relabel the maximal subtrees of \mathcal{T}_j so that $X_{\mathcal{T}_1^j}$ contains a fixed leaf. Let the maximal subtrees of \mathcal{T}_1^j be $A_{\mathcal{T}_1^j}, M_{\mathcal{T}_1^j}$, and relabel the maximal subtrees of \mathcal{T}_1^j so that $X_{A_{\mathcal{T}_1^j}}$ contains a fixed leaf. Suppose that \mathcal{T}_j displays exactly p_j distinct representations via the general switchings with the following property: for each reticulation vertex v in the associated basic cyclic pair of $N-(w_j, v_j, P_1^j, P_2^j)$, that is not a main two-way tree-path reticulation with respect to w_j , both v^{e_1} and v^{e_2} are in the general switchings. It is the case that p_j can either be 2^{k_j} or $2^{k_j} - 1$, where k_j is the number of main two-way tree-path reticulation with respect to w_j in the associated basic cyclic pair of $N-(w_j, v_j, P_1^j, P_2^j)$. If p_j is $2^{k_j} - 1$ then \mathcal{T}_j displays a representation twice, and the only way that \mathcal{T}_j can display a representation twice is when $(\mathcal{T}_2^j)^L \mid (X_{M_{\mathcal{T}_1^j}})^L = (M_{\mathcal{T}_1^j})^L$; hence w_j is replaced by $2^{k_j} - 1$ in COUNTINGLOCAL. If p_j is 2^{k_j} then $(\mathcal{T}_2^j)^L \mid (X_{M_{\mathcal{T}_1^j}})^L \neq (M_{\mathcal{T}_1^j})^L$, or at least one of $(X_{\mathcal{T}_2^j}), (X_{M_{\mathcal{T}_1^j}})$ contains a fixed leaf, in either case we see that w_j is replaced by 2^{k_j} in COUNTINGLOCAL. Since N is a locally independent network, we see that the number of distinct trees

displayed by N is $p_1 \times p_2 \times \dots \times p_n$. It follows that COUNTINGLOCAL returns the number of distinct trees displayed by N . The maximum number of cyclic pairs that need to be checked is less than or equal to the number of vertices in N . By Lemma 8.2, we have $|V| < 4|X|$. For a locally independent network, it takes linear-time to get the tree representing a basic cyclic pair; hence the running time of COUNTINGLOCAL is of $O(|X|^2)$. \square

Bibliography

- [1] C. Semple, M. Steel (2003). *Phylogenetics*. Oxford University Press.
- [2] P. Buneman (1971). The recovery of trees from measures of dissimilarity. In *Mathematics in Archaeological and Historical Sciences*, pp. 387–395.
- [3] C. Darwin (1859). *On the Origin of Species by Means of Natural Selection*. John Murray.
- [4] S. Clancy (2008). Genetic recombination. *Nature Educ.*, 1(1):40.
- [5] E. Koonin, K. Makarova, L. Aravind (2001). Horizontal Gene Transfer in Prokaryotes: Quantification and Classification. *Annu. Rev. Microbiol.*, 55:709–42.
- [6] R. Abbott, D. Albach, S. Ansell, J.W. Arntzen, S.J. Baird, N. Bierne, J. Boughman, A. Brelsford, C.A. Buerkle, R. Buggs, et al (2013). Hybridization and speciation. *J. Evolution. Bio.*, 26, 229–246.
- [7] T. Marcussen, S.R. Sandve, L. Sandve, M. Spannagl, M. Pfeifer, et al (2014). Ancient hybridizations among the ancestral genomes of bread wheat. *Science*, 345(6194), 1250092.
- [8] D.H. Huson, R. Rupp, C. Scornavacca (2010). *Phylogenetic Networks: Concepts, Algorithms and Applications*. Cambridge University Press.
- [9] A. Dress, K.T. Huber, J. Koolen, V. Moulton, A. Spillner (2012). *Basic Phylogenetic Combinatorics*. Cambridge University Press.
- [10] P. Lemey, M. Salemi, A. Vandamme (Eds.) (2009). *The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing*. Cambridge University Press.
- [11] E.O. Wiley, B.S. Lieberman (2011). *Phylogenetics: Theory and Practice of Phylogenetic Systematics*. Wiley-Blackwell.
- [12] M. Nei, S. Kumar (2000). *Molecular Evolution and Phylogenetics*. Oxford University Press.

- [13] L. Nakhleh, G. Jin, F. Zhao, J. Mellor-Crummey (2005). Reconstructing phylogenetic networks using maximum parsimony. In *Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference, CSB '05*, pp. 93–102.
- [14] L. van Iersel, C. Semple, M. Steel (2010). Locating a tree in a phylogenetic network. *Inform. Process. Lett.*, 110, 1037–1043.
- [15] M. Bordewich, C. Semple (2016). Reticulation-visible networks. *Adv. Appl. Math.*, 78, 114–141.
- [16] P. Gambette, A.D.M. Gunawan, A. Labarre, S. Vialette, L. Zhang (2015). Locating a tree in a phylogenetic network in quadratic time. In *Research in Computational Molecular Biology: 19th Annual International Conference, RECOMB 2015, Warsaw, Poland, April 12-15, 2015, Proceedings*, pp. 96–107.
- [17] A.D.M. Gunawan, B. Dasgupta, L. Zhang (2016). Locating a Tree in a Reticulation-Visible Network in Cubic Time. To appear in *Research in Computational Molecular Biology: 20th Annual International Conference, RECOMB 2016, Santa Monica, CA, USA, April 17-21, 2016, Proceedings*, doi: 10.1007/978-3-319-31957-5.
- [18] P. Gambette, A.D.M. Gunawan, A. Labarre, S. Vialette, L.X. Zhang (2015). Solving the tree containment problem for genetically stable networks in quadratic time. In *Combinatorial Algorithms: 26th International Workshop, IWOCA 2015, Verona, Italy, October 5-7, 2015, Revised Selected Papers*, pp. 197–208.
- [19] I.A. Kanj, L. Nakhleh, C. Than, G. Xia (2008). Seeing the trees and their branches in the network is hard. *Theor. Comput. Sci.*, 401, 153–164.
- [20] S. Linz, K. St. John, C. Semple (2013). Counting trees in a phylogenetic network is #P-complete. *SIAM J. Comput.*, 42, 1768–1776.
- [21] M. Fischer, L. van Iersel, S. Kelk, C. Scornavacca (2015). On computing the maximum parsimony score of a phylogenetic network. *SIAM J. Discrete Math. (SIDMA)*, 29(1):559–585.
- [22] P. Cordue, S. Linz, C. Semple (2014). Phylogenetic networks that display a tree twice. *B. Math. Biol.*, 76, 2664–2679.
- [23] S.J. Willson (2012). Tree-average distances on certain phylogenetic networks have their weights uniquely determined. *Algorithm. Mol. Biol.*, 7:13.

- [24] G. Cardona, F. Rossello, G. Valiente (2009). Comparison of tree-child phylogenetic networks. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 6, 552–569.
- [25] G. Cardona, M. Llabrés, F. Rosselló, G. Valiente (2008). A distance metric for a class of tree-sibling phylogenetic networks. *Bioinformatics*, 24, 1481–1488.
- [26] C. McDiarmid, C. Semple, D. Welsh (2015). Counting phylogenetic networks. *Ann. Comb.*, 19, 205–224.
- [27] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein (2001). *Introduction to Algorithms*. MIT Press and McGraw-Hill.
- [28] S.J. Willson (2010). Properties of normal phylogenetic networks. *B. Math. Biol.*, 72, 340–358.
- [29] A.V. Aho, J.E. Hopcroft, J.D. Ullman (1974). *The Design and Analysis of Computer Algorithms*, Addison-Wesley.
- [30] S. Buss (1997). Alogtime algorithms for tree isomorphism, comparison, and canonization. In *Computational Logic and Proof Theory: 5th Kurt Gödel Colloquium, KGC '97 Vienna, Austria, August 25-29, 1997, Proceedings*, pp. 18–33.